Flexible Robot Platform For Autonomous Research

By

David John Hall (BComp)

A dissertation submitted to the
School of Computing
in partial fulfilment of the requirements for the degree of

Bachelor of Computing with Honours

University of Tasmania

November 2005

Declaration

I, David Hall declare that this thesis contains no material, which has been accepted for the award of any other degree or diploma in any tertiary institution. To my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Signed: Date:

David Hall

Abstract

The field of mobile robotics is receiving increasing levels of research. However, the simulation tools which are utilised in the creation of new mobile robot algorithms can produce algorithms which do not work in the real world. In order to try and minimise this problem a flexible robot platform has been created which allows the testing of a variety of algorithms. The platform facilitates the testing of algorithms normally only simulated by allowing algorithms to be easily tested in the real world. Utilising the flexible robot platform for testing algorithms allows higher quality research, as algorithms can be assessed with physical evidence.

Acknowledgements

I would like to thank the following people for their help throughout the year:

My supervisors Dr Daniel Rolf and Dr Waheed Hugrass for all their patience, support and advice.

Mr Tony Gray for his assistance in ordering and collecting all of the parts for the project.

Andrew Spilling for his repeated help booking the seminar room.

My friend James for his ongoing suggestions and ideas.

My family for their continued support, assistance and encouragement.

Table of Contents

List of Figures

List of Tables

List of Equations

Chapter 1 Introduction

1.1 Background

Recently the field of mobile robotics has become a focus of a great deal of research, with many academic institutions worldwide developing new algorithms and technologies. With the volumes of research being undertaken in the field, it is becoming increasingly popular to develop and test new algorithms for mobile robots using simulation tools. They provide researchers with many plausible and enticing benefits, such as low cost and a high speed of development. They can simply test their algorithms on a regular computer and gather results.

Simulation has many benefits, however, they struggle to model the real world accurately and as a consequence algorithms which work well in a simulation, completely fail when applied in the real world. Implementing algorithms on real world mobile robots has been found to be very successful, providing an excellent indication of the success or validity of an algorithm. Traditionally testing robots in the real world has involved a process of creating a mobile robot to test a specific algorithm. The process of building the new robot and then testing an algorithm can be an extremely time consuming process. For example, testing an autonomous underwater vehicle requires a very long period for testing, as the robot must be tested in water it becomes difficult to access and monitor during experiments. Constructing a mobile robot can also be very expensive and if the robot is too inflexible to be used for other projects after a certain algorithm has been tested, building the mobile robot often cannot be justified.

This results in many new mobile robot algorithms never being tested in the real world. However with these algorithms only being simulated, whether the algorithms actually work in the real world cannot be guaranteed or relied on. If these algorithms could be tested in the real world in a relatively short time and with a low cost the quality of the algorithm being researched would be greatly increased.

1.2 Hypothesis

It would be beneficial to mobile robot research if more algorithms could be tested in the real world. It is thought that a flexible robot platform, designed to be used for different applications, would allow more mobile robot algorithms to be tested in the real world. In an attempt to minimize the difficulties in testing mobile robot algorithms in the real world, this thesis aims to investigate the hypothesis:

That it is possible to create a mobile robot platform, with the flexibility to test a variety of algorithms.

Stemming from this hypothesis this thesis also begins to investigate the possibility of utilizing a flexible mobile robot to implement algorithms for different kinds of mobile robot platforms.

1.3 Approach

In the investigation of the hypothesis, Chapter 2 provides an overview of previous research into mobile robot flexibility. A mobile robot platform referred to as the Flexible Robot Platform or FRP has been constructed and detailed in Chapter 3, with the aim to possess the highest degree of flexibility possible. Chapter 4 details the testing and analysis of the Flexible Robot Platform. This provides the results necessary for the evaluation of the hypothesis in Chapter 5.

Chapter 2 Literature Review

2.1 Introduction

The following literature review aims to provide an introduction to robots and more specifically mobile robots and the sensors they commonly use. Also to examine previous research into the inclusion of flexibility in robotic platforms, an analysis of terrestrial mobile robots commonly used for research and an examination of the methods used to test algorithms for mobile robots. The Encyclopaedia Britannica describes a robot as;

"Any automatically operated machine that replaces human effort, though it may not look much like a human being or function in a humanlike manner"(2005).

This description fits all robots ranging from an automatic conveyor belt, such as those found at supermarkets, to a fully autonomous robot from a functional perspective. Physically robots are a collection of sensors, processors and manipulators which allow a robot to perform its function. Systematically, sensors provide input to a robot about both itself and its environment. Processing determines what to output and manipulators provide the robot's output.

Robots can be broadly divided into two main groups or types, stationary robots and mobile robots. Stationary robots are robots which operate in a highly structured environment, and thus can be easily programmed to perform a task within that environment. For example, probably the most common stationary robots are those used in manufacturing, such as robotic arms which only need to operate within the environment of the factory or warehouse where they are located. Mobile robots, however, do not operate in a static or pre-determined environment. As a result of this they must be able to handle changes in their environment and adapt or cope with situations that may not have been specifically accounted for. This has limited the development of mobile robots as the computing power has not always been available to implement complex algorithms which can adapt to new or changing environments.

2.2 *Mobile Robots*

There are several different kinds of mobile robot platforms. Many mobile robot platforms are similar to existing vehicles such as cars, boats or aeroplanes. However, mobile robots also take on different or innovative forms which would not be considered for use for human travel. Mobile robots can be categorised into the following (Dudek & Jenkin 2000):

2.2.1 Airborne

Airborne robots usually take the form of traditional aircraft or helicopters, however, they may also take other forms including parachutes, dirigibles or even rockets (for example a cruise missile). Flying robots usually require constant momentum or drive to remain airborne, which affects the way that they are controlled. Airborne platforms share some control properties with aquatic platforms due to the fact that their locomotion is not based on contact with stationary objects or terrain.

2.2.2 Aquatic

Aquatic robots operate either in or on water. Aquatic robots are more commonly submarine, although surface robots are still used. Aquatic robots have mainly been used for marine research to access underwater areas that are not easily reached by humans. For example, robotic submarines or autonomous underwater vehicles can endure deep water missions for much longer than manned submarines.

2.2.3 Space

Robots in space are unique in that they must operate in zero or microgravity. Also, their form is affected less by the environment and more by the robot's function. However, space robots often need to be small and light to reduce the cost of getting them into orbit. A satellite which can automatically hold or change position is an example of a mobile robot.

2.2.4 Terrestrial

Terrestrial robots are by far the most common form of mobile robots as they are easiest to build and operate. Simpler wheeled terrestrial robots are more common, however, the use of legged mobile robots is becoming more common with improvements in technology. Terrestrial robots also take the form of everyday vehicles such as cars and tracked vehicles like excavators.

2.3 Terrestrial Mobile Robot Sensors

Mobile robots make use of a variety of different sensors in order to derive information about themselves and their environment. Sensors which are commonly used by terrestrial mobile robots have been examined.

2.3.1 Tactile Sensors

Figure 2.1 Switch, strain gauge and piezoelectric sensors

Tactile sensors allow the detection of physical objects. The simplest of tactile sensors simply close a switch when contact is made. More sophisticated sensors provide information on the strength or amplitude of contact. Common sophisticated tactile sensors consist of strain gauges which vary output depending on the level of deformation of the sensor and piezoelectric transducers which provide varied voltage bursts depending on the deformation. (Nehmzow 2003, pp. 26-7)

2.3.2 Ultrasonic Sensors

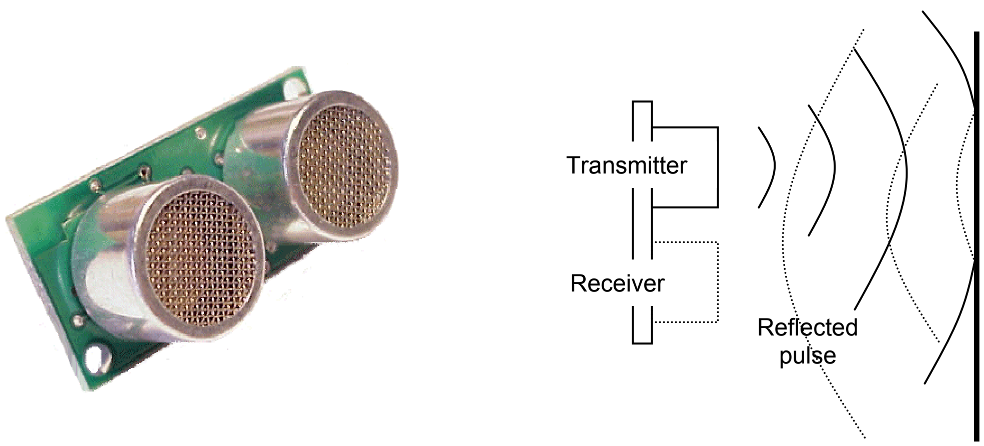


Figure 2.2 SRF04 Ultrasonic rangefinder module and diagram of operation

An ultrasonic sensor is made up of one or more transducers and some control circuitry. The sensor can detect whether there is an object in front of it and how far away it is. The sensor works by sending out a high frequency ‘ping’ and then listening for the reflection of the ‘ping’ with the same or another transducer. The longer the ‘ping’ takes to return to the sensor the further away the object is assumed to be. (McComb 2001, p. 633)

2.3.3 Infrared Sensors

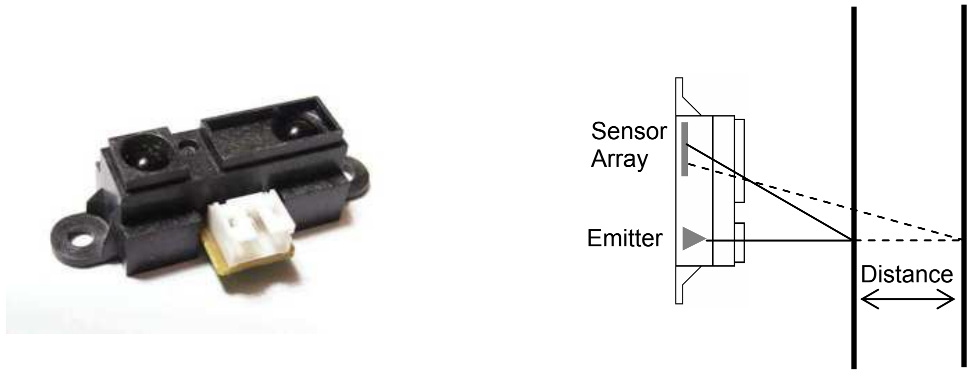


Figure 2.3 GP2D12 Infrared rangefinder and diagram of operation

Infrared sensors consist of an Infrared (IR) emitter (usually an IR Light Emitting Diode) and one or more IR sensors. Simple IR sensors simply register if the IR signal is detected, however IR rangefinders utilise an array of IR sensors. The distance of an object from the sensor can be determined through triangulation by sensing which element of the array is illuminated with reflected IR light. Different elements of the IR sensor array are illuminated due to the changing angle of the reflected light with distance from the sensor module. (Nehmzow 2003, pp. 27-8)

2.3.4 Laser Sensors

Laser rangefinders can detect the distance of objects using three methods. These methods are triangulation, time-of-flight and phase-based. Triangulation works in the same way as the infrared sensor, however makes use of laser light. Time-of-flight works on the same principles as ultrasonic sensors, determining distance from the time taken for the signal to return. Phased-based rangefinders determine the distance of an object by changes in the phase of the reflected light. Laser sensors are also referred to as LIDAR (light detection and ranging). (Dudek & Jenkin 2000, pp. 67-8)

2.3.5 Digital Compass

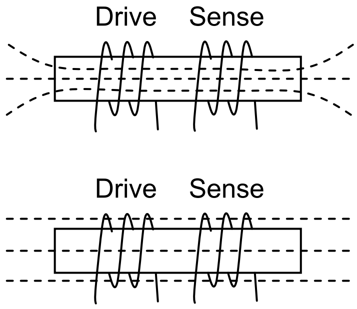


Figure 2.4 CMPS03 Digital Compass and flux in unsaturated (top) and saturated (bottom) core

A digital compass measures the horizontal component of the earth's magnetic field. There are five main compass technologies utilised in robotics: Mechanical, Fluxgate, Hall-effect, Magneto-resistive and Magneto-elastic. The most commonly used are Fluxgate compasses which measure the earth's magnetic field using a controlled electromagnet. A Fluxgate compass consists of a drive and sensing coil on a common core. By alternately driving the drive coil (altering the flux through the core) a voltage is induced in the sensing coil which varies depending on the ambient magnetic field. Two cores are required to sense north. (Borenstein, J, Everett & Feng 1996; Nehmzow 2003)

2.3.6 Encoders

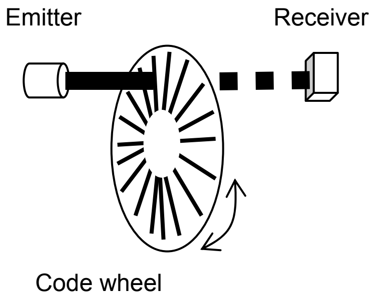


Figure 2.5 HEDS-5500 Optical Encoder and diagram of operation

Encoders determine the amount of revolution of a shaft by having a disc with a specific pattern (code wheel) with which a sensor produces a signal of pulses for each part rotation of the shaft. The amount of rotation in the shaft can be determined by counting the number of pulses. The sensor for the pattern on the disc is usually a light sensor, however contact or hall-effect sensors can also be used. (Borenstein, J, Everett & Feng 1996, pp. 13-7)

2.4 Mobile Robot Flexibility

Recently the creation of mobile robot platforms which are flexible has become increasingly desirable, due to its advantages to aspects of development such as lowering cost and shortening implementation times, as well as becoming increasingly viable due to new technologies and cheaper, more powerful components. Previous research into creating flexible robot platforms has mainly had flexibility as a secondary goal for the platform being developed, however recent research has had a greater focus on the flexibility of mobile robot platforms.

Gerecke and others required a robotic platform which was flexible enough for use in teaching and research endeavours. In order to try and achieve this goal they are developing a mobile robot platform (MoRob) and have identified several requirements for their platform to be successful. These include a comprehensive application program interface (API), interfaces and libraries, a variety of modules for sensing and control and comprehensive documentation (Gerecke, Hohmann & Wagner 2003; Wagner et al. 2004).

A researcher at the Fraunhofer Institute for Autonomous learning has developed a modular drive system for a mobile robot platform which allows the robot to be changed between an omni-directional and differential drive system. The system is made highly versatile due to this modularity and increases the variety of tasks which it can undertake (Bose 2004).

Researchers who have developed an autonomous two wheel drive tractor have found that by allowing their platform to accept new sensors and using a relatively powerful computer for processing their platform has been highly flexible. Being able to support a number of concurrent and diverse research projects (Will et al. 1998).

Another robot platform developed by researchers for personal robotics called MILO has also been designed for flexibility and has taken advantage of off-the-shelf components to provide the platform with a level of modularity. Some of their aims during the creation of the platform were cost effectiveness, reliability, safety, efficiency, flexibility, expandability and ease of programming (Salemi et al. 2005).

Also in order to develop a controller for creating robots with the flexibility needed for education and research, Loose and others have created a robot controller (RCUBE). The unit is modular and allows the connection of sensors and actuators, as well as being cost effective and providing sufficient processing power for various applications (2004).

Flexibility has also formed an essential aspect to enable a variety of research directions for a set of low cost robots called CotsBots developed at the Berkeley Sensor and Actuator Center. The CotsBots achieve flexibility through the ability to interface with new sensors, and TinyOS an open source, modular operating system with a CotsBots API (Bergbreiter & Pister 2003).

From the research examined it is possible to identify some commonly cited factors which have made the platform under development by a particular researcher more flexible. All of the platforms developed have the ability to easily add new sensors to the platform. The majority of these platforms also have a developed API and low cost. Other less cited factors were detailed documentation, re-configurable chassis and the use of off-the-shelf components.

2.5 Terrestrial Mobile Robot Platforms

In addition to the research into the area of mobile robot flexibility, there are a number of commercial mobile robot platforms available, which also exhibit varying levels of flexibility. Many of these platforms are used by academic institutions for teaching and research. The most common of these platforms have been examined in order to provide a complete picture of the current state of flexibility in mobile robot platforms.

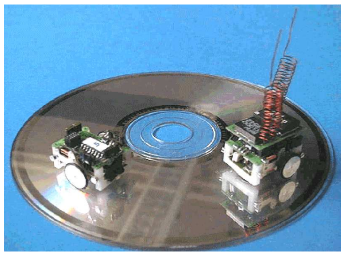
2.5.1 ActivMedia Robotics
Pioneer 1

2.5.2 Evolution Robotics
ER1 Personal Robot System

2.5.3 K-Team Corporation
Khepera II

2.5.4 Lego™ MindStorms

2.5.5 Nomadic
Technologies Super Scout II



2.5.6 Micro Robot “Alice”

Figure 2.6 Terrestrial mobile robot platforms

2.5.1 ActivMedia Robotics Pioneer 1

2.5.1.1 About the platform

The Pioneer 1 is a robot platform produced by ActivMedia Robotics. The Pioneer 1 robot has been very popular for use in research at all levels and has several new versions and variations. The platform is relatively expensive costing around three thousand dollars (*Outfitting A Robot Laboratory*). The Pioneer 1 has a differential drive system with a real wheel for stability. The platform also incorporates 7 ultrasonic sensors in an array on the front of the robot for environment sensing. The platform can also interface with additional sensors and hardware such as a laptop for increased functionality. In addition the Pioneer 1 platform has an extensively developed API which allows the robot to be easily programmed and also simulated for program development. (*ActivMedia Robotics*)

2.5.1.2 Platform use

The Platform has been used in an undergraduate research project at the University of North Dakota with the task of serving food at the association for Artificial

Intelligence's annual robot competition. The robot platform had to be programmed to handle problems such as crowd navigation (Maxwell & Meeden 2000). Also, the platform has been used in more cutting edge research by Thrun and others for 3D real-time mapping with multiple robots (2000). Ibeanusi and others have also used the Pioneer 1 platform to research dead reckoning and the extent to which sonar can be used to increase the robot's navigational accuracy (1999).

2.5.2 Evolution Robotics ER1 Personal Robot System

2.5.2.1 About the platform

The ER1 Personal Robot System (ER1) is produced by Evolution Robotics. The platform is moderately priced at around seven hundred dollars, not including an additional laptop that is required by the platform. The ER1 utilises a differential drive system for locomotion and uses a webcam for navigation. An API allows control of the robot and also includes functionality for vision based object recognition and navigation. Analogue and digital input/output (I/O) lines are also provided for interfacing additional hardware. The main feature of this platform is its reconfigurable chassis which allows the robot to be reconfigured for a particular application. (*ER1 Personal Robot System* 2005)

2.5.2.2 Platform use

The ER1 robot has been used by researchers at the University of Georgia for research into the development of robotic wheelchairs. An advantage provided by the ER1 was that the chassis could be configured to roughly match that of an electric wheel chair (Ono, Uchiyama & Potter 2004).While the platform is more powerful due to the use of a laptop for processing and control, it has been found to have limited sensing capabilities and a lack of flexibility within its API (Gerecke, Hohmann & Wagner 2003).

2.5.3 K-Team Corporation Khepera II

2.5.3.1 About the platform

The Khepera II is a small robot platform made by K-Team Corporation and is sold specifically for research and educational use. The platform is quite expensive at around two thousand five hundred dollars. The Khepera II is 7cm in diameter and has a differential drive system. Also included are 8 built in infrared sensors and an onboard processor for control. Three analogue inputs are also provided for additional sensors as well as stackable boards to enhance functionality. In addition K-Team provides an extensive API and simulation tools for the robot platform. (*K-Team Corporation* 2005)

2.5.3.2 Platform use

The platform's small size makes it ideal for use on a desk top and it is sophisticated enough for many research tasks. The platform has been used in an undergraduate curriculum at California State University USA as part of research into using robots as a learning tool (Challinger 2005). Dozier has also used the Khepera II robot platform to implement a new method of teaching neural networks (Dozier 2001).

2.5.4 Lego™ MindStorms

2.5.4.1 About the platform

Mindstorms™ is a robotics kit produced by Lego™ (*Lego(tm) Mindstorms(tm)* 2005) and is popular due to its relatively low cost of around three hundred dollars. The platform is structured around an Robotic Command Explorer (RCX) 'brick' controller which contains a programmable controller and battery unit. The relatively small RCX unit can have Lego™ pieces attached to it to construct a robot of any desired chassis configuration. The RCX supports three inputs and three outputs, accepting three proprietary Lego™ sensors, a light sensor, temperature sensor and touch switch. The RCX outputs can be used to control Lego™ motors.

2.5.4.2 Platform use

The two main advantages of the Mindstorms robot platform are its low cost and re-configurability. However, the platform is restricted by the low processing capabilities of its controller, the inability to interface third party hardware and the limited strength of the platform for carrying additional loads. These constraints have meant that the platform has been mostly limited to teaching applications. Vamplew has implemented reinforcement learning algorithms using the platform despite its limitations (2004). Also, Fagin and Merkle have used the Mindstorms platform to perform a year long study into the effectiveness of robots in teaching computer science (2003).

2.5.5 Nomadic Technologies Super Scout II

2.5.5.1 About the platform

The Super Scout II robot platform was sold commercially by Nomadic Technologies. Although the company has ceased operations, the platform is still used by various organisations for research. There is now an open source project (Sprouse) supporting the robot platform. The Super Scout II is a differential drive robot platform with 16 sonar sensors mounted in an array over 360 degrees around the robot. The robot has an onboard computer with hard drive for control and has an API which can be programmed using the C or C++ programming languages.

2.5.5.2 Platform use

The platform has been used by the University of North Dakota for their entry in the American Association for Artificial Intelligence's annual robot competition, undertaking navigational and recognition tasks (Maxwell & Meeden 2000). The platform has a greater processing ability than most platforms which makes it able to undertake a greater variety of algorithms. The Super Scout II has also been used more recently for cooperative soccer playing robots using artificial intelligence (Lima & Custódio 2004).

2.5.6 Micro Robot Alice

2.5.6.1 About the platform

Alice is a micro robot developed at the Swiss Federal Institute of technology Lausanne for research and education purposes (Caprari et al. 1998).The Alice robot is available commercially from K-Team Corporation, for around six hundred dollars (*K-Team Corporation* 2005). The most unique feature of the platform is its unique size of approximately 2x2x2cm. The platform also has 4 infrared proximity sensors and a small 4 MHz microcontroller. Also a radio module can be added to the platform for control from a PC.

2.5.6.2 Platform use

The platform's small size makes it ideal for use on a laboratory work bench, making it easy to test algorithms. However, the platform's size is also very limiting, giving it limited scope for the attachment of different sensors and larger processing units. The main advantages of the Alice robot are that it is low cost, has a low power consumption and some ability for the addition of small modules (Caprari et al. 1998).

2.6 Robot Algorithm Testing

There are three main methods used for the testing of robotic algorithms. These are real world, simulated and hybrid testing. Each method has certain strengths and weaknesses which are examined further.

2.6.1 Real World

Real world testing involves implementing an algorithm on a physical robot platform such as those in Figure 2.6. The success of the algorithm can then be evaluated by seeing how it performs on the robot in the real world. The advantage of real world testing is that an algorithm can be confirmed to work. The disadvantages of real world testing are that it can be time consuming to test algorithms and costly (Michel 2004).

2.6.2 Simulated

Simulation involves simulating a robot platform in a virtual world to test algorithms. The advantages of this are that algorithms can be quickly tested (faster than real time) and cost less than real robots (Lee, Nehmzow & Hubbold 1998; Michel 2004). However, the main disadvantage of simulation is that due to the fact that simulators cannot model the real world exactly, algorithms which work in simulations can completely fail in the real world (Brooks 1992). This has limited the effectiveness of simulations (Lee, Nehmzow & Hubbold 1998).

2.6.3 Hybrid

Hybrid testing involves using a combination of simulation and real world testing. This approach attempts to gain the best of both approaches, however the inclusion of real world systems makes hybrid testing equally as slow. Hybrid testing has been successful in testing mobile robots before using them in the real world (Kuroda, Aramaki & Ura 1996).

Chapter 3 Methodology

This chapter details the process undertaken to create the Flexible Robot Platform (FRP). This process consists of two stages, the examination of existing platforms to determine desirable characteristics for a flexible mobile robot platform and the implementation of a mobile robot platform (the FRP) which possesses as many of the identified characteristics as possible.

3.1 Existing Mobile Robot Platforms

In order to construct a flexible mobile robot platform, the features or factors which make a mobile robot flexible needed to be determined. Previous research (see section 2.4) has identified several factors which increase the flexibility of a mobile robot platform. The factors are summarised as follows:

- Ability to interface 3rd party components (sensors & actuators)
- Well developed and documented API
- Re-configurable chassis
- Low cost
- Modularity

In order to identify additional factors which create flexibility within a mobile robot platform, an analysis of the commercially available mobile robot platforms was performed. Table 1 shows the advantages and disadvantages discovered for each platform with respect to flexibility.

Mobile Robot Platform	Advantages	Disadvantages
ActivMedia Robotics Pioneer 1	• Well Developed API • Up to 23 Kg payload • Ability to carry additional processing • Ability to interface third party components	• Fixed sensors • Fixed components and internal electronics
Evolution Robotics ER-1 Personal Robot System	• Re-configurable chassis • Re-positionable sensors	• Limited API • No ability to interface third party components
K-Team Khepera II	• Laboratory or work bench operation	• Limited onboard processing • No ability to interface third party components
Lego Mindstorms	• Re-configurable chassis • Re-position able sensors • Low Cost	• Limited onboard processing • No ability to interface third party components
Nomad Super Scout II	• Ability to carry additional processing • Ability to interface third party components	• Fixed sensors • Fixed components and internal electronics
Micro Robot "Alice"	• Work Bench operation	• Very limited onboard processing

Table 1 Mobile robot platform advantages and disadvantages

From the factors in Table 1 and those identified from previous research, a new summary of the factors of flexibility in mobile robots has been produced (Figure 3.1). The modularity of a mobile robot platform was not included as it was deemed to be an inherent property of other factors.

- Ability to carry additional processing
- Ability to interface third party components
- Laboratory or work bench operation
- Large payload capacity
- Low cost
- Re-configurable chassis
- Re-positionable sensors
- Well developed and documented API

Figure 3.1 Flexibility factors in mobile robots

The flexibility factors in Figure 3.1 provide a set of requirements which have been used to create the Flexible Robot Platform's design. The factors also serve as a basis to analyse the flexibility of the created platform (FRP) and other mobile robot platforms.

When attempting to create a mobile robot platform with all of the factors in Figure 3.1 the factors can conflict with each other. This occurs due to the counter-productive nature of some of these factors. For example a robot which has an extremely configurable chassis may as a result not be able to carry a particularly large payload due to the chassis of the robot having less structural integrity. This means that when determining the final design of a mobile robot platform, some compromises may need to be made to ensure a platform is as flexible as possible without unduly impairing the other aspects of the robot platform. One factor which has an overbearing influence on the ability to include other factors into a mobile robot design is low cost. By limiting the cost of a mobile robot design it may not be possible to satisfy all of the factors to create a flexible robot platform. For example an entirely re-configurable chassis may cost considerably more than one that is only partially configurable. As a result compromises may need to be made to designs in order to create a balance amongst all the flexibility factors.

3.2 Implementation

3.2.1 Design

Having thoroughly investigated the designs of existing mobile robot platforms, the design for the 'flexible robot platform' was produced. The platform needed to be as flexible as possible within the budget and time constraints. The project had a small budget of around one thousand five hundred dollars and was to be constructed and tested over a period of 2 semesters (9 months). Initially existing commercially available robot platforms such as the Pioneer robots from ActivMedia Robotics (*ActivMedia Robotics*) were investigated. However these were found to be unsuitable for two reasons. Firstly, the platforms sensors are in fixed positions making the platforms too inflexible for the project. Secondly, the platforms are relatively expensive and would not allow the developed platform to have a low cost.

For this reason and to maintain the modularity of the flexible robot platform the use of mostly pre-constructed or off-the-shelf sub-components was selected. The use of sub-components allowed the utilisation of already developed and supported components and meant that the platform could be more easily constructed within the projects timeframe.

The design of the flexible robot platform is centred around the electrical and data connections of the components, as the platform allows for the component's physical configuration to be changed into any desired state. Each of the components selected for the flexible robot platform are shown in Figure 3.2 and are explained in detail in section 3.2.2.

3.2.2 Components

The Flexible Robot Platform's components can be divided into three main categories, chassis and power, processing and control and sensors. Figure 3.2 shows how each of the components are connected with the FRP.

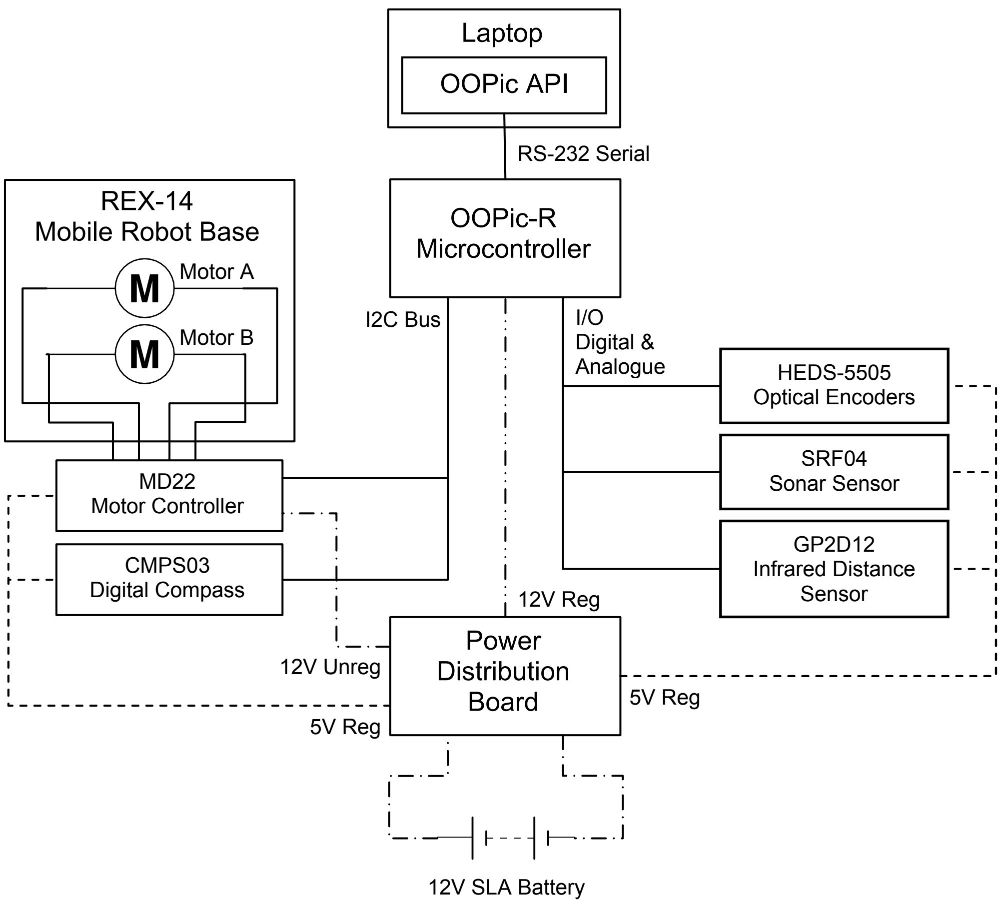


Figure 3.2 Flexible Robot Platform Block Diagram

3.2.2.1 Chassis and Power

REX-14 Mobile Robot Base

The REX-14 Mobile robot base from Zagros Robotics (*Zagros Robotics*) was selected as it is pre-constructed, saving development time and is a 'bare bones' base meaning that minimal sensors were already installed. The platform is a 35x35cm square platform with two geared motors and wheels which form a differential drive system. The platform also has installed an encoder on each gearbox. A Multi-Degree of Freedom drive system would have been preferred for the platform as it would allow the platform to model each of the terrestrial mobile robot drive systems. However, multi-degree of freedom drive systems are not commercially available and would have been too costly to produce within the projects time frame. The flexibility of the platform is slightly reduced due to this compromise.

12V SLA Battery

A 12V Sealed Lead Acid battery was chosen to power the robot. This was determined to be most suitable as the motors on the REX-14 part of the platform require a 12 volt supply. This allows motor power to be taken directly from the battery. Also any lower voltages required by the platform could be regulated down from the battery. Lithium-ion batteries were also considered for the platform, however were not as cost effective as the lead acid based batteries and did not provide any significant benefits in terms of flexibility. The battery in the FRP can be easily removed, allowing new or different batteries to be installed very easily.

Power Distribution Board

As some of the components chosen for the platform require a 5 volt power supply and are more sensitive, a power distribution board was designed to distribute the supply of power from the onboard battery. The distribution board provides a regulated 5 and 12 volt supply, as well as a method of easily connecting and disconnecting the robots components to the power supply. A data sheet showing the design of the power board is included in Appendix D.

3.2.2.2 Processing and Control

Laptop / Portable Computer

The design of the flexible robot platform makes use of a laptop computer for higher level processing and control. A laptop computer was chosen as they are widely available and are easy to use. It is also possible that a different processor be used in place of a laptop, such as a small form factor personal computer or other processor. The software which runs on the laptop to control the FRP is detailed in section 3.2.4.

OOPic-R Microcontroller

The OOPic-R microcontroller from Savage Innovations was chosen as it can provide the FRP with a significant flexibility for a low cost. The OOPic-R is designed specifically for robotics applications and provides an interface to any desired components (Savage Innovations 2005). One advantage of this controller over other microcontrollers is that it comes with firmware which provides interfaces for many popular sensors and actuators. The micro controller also has the ability to be controlled via a serial port, which means that the laptop can control devices 'through' the OOPic. The OOPic also provides an I2C interface which is a bus system developed by Phillips for integrated circuit intercommunication (Philips Electronics 2005). Many third party components operate using the I2C bus system. The OOPic controller is examined in closer detail in section 3.2.4

MD22 H Bridge Motor Driver

The MD22 Motor Driver is a generic motor controller which can control two motors up to 50 volts at 5Amps. This not only meets the requirements of the motors on the REX-14 base, but also leaves sufficient margin to use different motors with the platform. The motor controller is also flexible in the methods which can control it, allowing control via servo motor signals, analogue voltages and an I2C bus.

3.2.2.3 Sensors

A variety of sensors were chosen to use with the mobile robot platform to test the flexibility to integrate with different components and for use when testing the flexible robot platform.

SRF04 Sonar Sensor

The SRF04 sonar ranger (Figure 2.2) is a dual transducer ultrasonic range finder. The SRF04 is still relatively small despite the dual transducers and is operated by a simple trigger and return pulse. The main advantage of the SRF04 over other available sonar ranging modules is that the SRF04 has a range from 3cm to 3m. Single transducer sonar modules typically only have a range of 15cm to 6m and although they have a longer range they cannot detect very close objects which are more important to a mobile robot platform.(Devantech Ltd 2003a)

HEDS-5505 Optical Encoders

Two HEDS-5505 two channel optical encoders (Figure 2.5) manufactured by Agilent Technologies (Agilent Technologies 2001) are used to provide internal sensing of wheel rotation. The encoders provide both a measure of the rotation and the direction of rotation. The encoders have a resolution of 500 counts per revolution, which is the same as the Pioneer series robots from ActivMedia Robotics (*ActivMedia Robotics*). The encoders were pre-installed into the REX-14 Base motor gearboxes, which allowed construction time to be minimised.

GP2D12 Infrared Distance Sensor

A Sharp GP2D12 Infrared Distance Sensor (Sharp Electronics Corp.) was chosen as they are commonly used in robotics. They provide a short range distance measurement (<80cm) which is suitable for obstacle detection. The sensor is pictured in Figure 2.3 and its basic operation is described in section 2.3.3.

CMPS03 Digital Compass

The CMPS03 digital compass (Figure 2.4) provides a direction heading of north to the robot platform (Devantech Ltd 2003b). The compass has an accuracy of 3-4° and aids navigation of the FRP.

3.2.3 Construction

During the construction of the flexible robot platform several techniques were used to increase the flexibility of the platform. These techniques are examined in further detail below.

3.2.3.1 Component Connection

Traditionally the connection of components may have taken place by soldering required connections onto controllers and power supplies or by connecting them together with large plugs. However, this approach has the main disadvantage that it makes it harder to change the configuration of the robot, as if a sensor needs to be moved or removed it needs to be unsoldered or removed from a plug. Another approach which avoids this problem is to use solder-less bread boards to connect components. These work well for rapid prototyping of a circuit, however are not very robust and may become dislodged on a travelling robot.

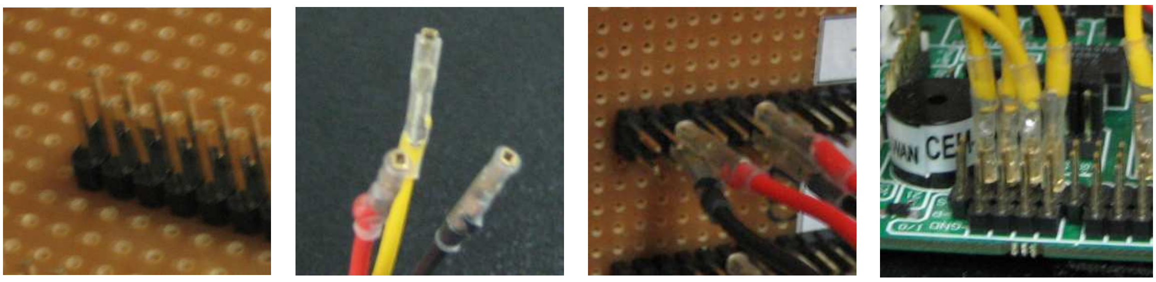


Figure 3.3 FRP Component connections

To avoid the time consuming process of soldering and the frailty of bread boards, all connections for the components are connected using a pin and socket system. This is advantageous as the connections are easily changed and are sturdy enough to stay connected during robot travel. The OOPic controller has pins for component connections and so fits well into this system. The power distribution board was also designed to allow pin-socket connections. However, the pin-socket method of connection is unsuitable for the high currents required to drive the motors on the REX-14 Base. In this case, screw terminal connections were used to still allow relatively simple connection of the motors and other high current components. The overall solution provides a system where it is very simple and quick to alter the configuration of connections within the platform.

3.2.3.2 Sensor Mounting

Sensor mounting is an important aspect of a robot's design. The mounting location of a sensor determines the meaning of the sensor's output for the robot as a whole. For example, a sensor mounted looking forward might detect an object in a robot's path, whereas a sensor mounted looking to the left may detect objects the robot is passing. Obviously the object detected directly in the robot's path has much greater implications to the robot as it will need to take some action to avoid the obstacle. The majority of robot platforms have sensors installed into the robot's base in an arbitrary manner usually forming some kind of sensor array. This may be suitable most of the time however is limiting should the sensor configuration need to be changed, especially if the mounting is part of the robot's chassis.

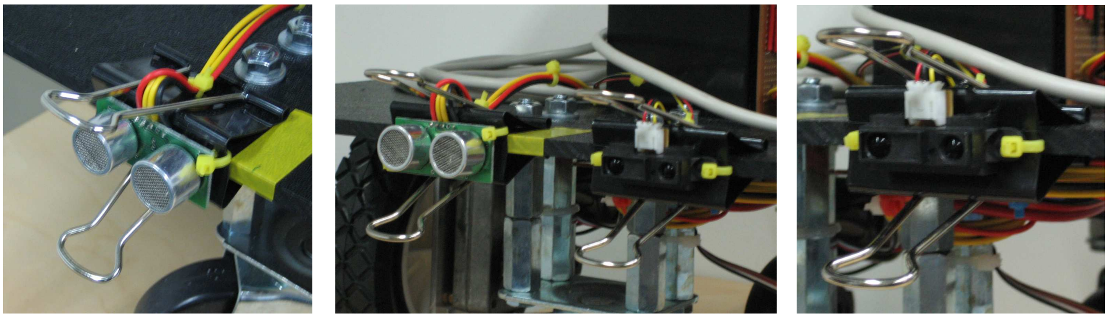


Figure 3.4 Flexible Robot Platform Sensor Mounting

To overcome this limitation, sensors for the flexible robot platform have been mounted on 50mm fold back clips. This allows the sensors to be firmly clipped to virtually any part of the robot's chassis. Should a sensor then need to be moved, removed or replaced it can simply be unclipped and unplugged. The clips also provide a small amount of protection to the sensor from accidental collision. This provides the FRP with a re-positionable sensor system, allowing sensors to be placed in any desired configuration.

3.2.3.3 Component Mounting

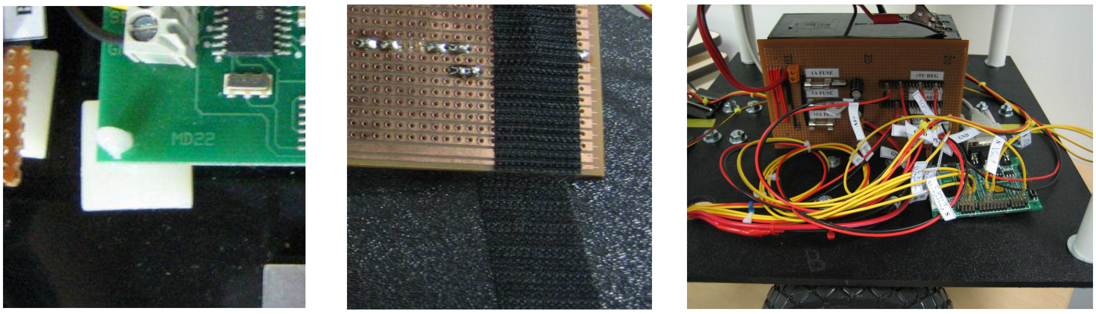


Figure 3.5 Flexible Robot Platform Component Mounting

In order to preserve the flexibility to move components within the platform, the non sensor components have been attached to the robot's chassis using Velcro™ and printed circuit board (PCB) Standoffs. This allows the components on the FRP to be moved or re-positioned as required. However, the components are also adequately attached to the platform so that they do not fall off during travel.

3.2.4 OOPic API

In order to control the FRP's sensors and actuators the laptop must communicate with the OOPic Controller within the platform. This is a complex process, so in order to simplify the control of the platform an API for the OOPic microcontroller was created. The API for control provides the FRP with a great deal of flexibility as it allows algorithms for the FRP to be quickly and easily implemented. In addition, a programmer of an algorithm only needs a basic understanding of the functioning of the components being controlled.

3.2.4.1 OOPic Operation

The OOPic microcontroller (Savage Innovations 2005) operates differently from a traditional microcontroller. Rather than providing simple access to digital and analogue I/O lines, the OOPic controller provides firmware objects. These objects allow both simple I/O access and more advanced access with objects which operate specific devices attached to the controllers I/O. In order to setup these devices the OOPic is programmed with a definition of the objects required; optionally a user can place an entire running control program onto the OOPic. However, the OOPic has limited space for storing programs.

Each of the objects has an associated memory space with object properties which can be read and written to control the device that the object is connected to. In order to access this memory from a laptop the OOPic's serial control protocol (SCP) is used. The SCP allows parts of memory to be accessed on the OOPic controller and also the ability to transfer data over the OOPic's I2C Bus, allowing access to any device connected.

3.2.4.2 API Design

The API has been designed to leverage the features provided by the OOPic controller. The API models the memory of OOPic objects and registers of I2C devices with objects in the API and provides methods to manipulate them. In order to make the API and algorithms using it as fast as possible a local copy of the OOPic memory and I2C object registers are stored in local memory (on the laptop). This means that the OOPic only needs to be accessed to acquire new data or write new command values.

In order to make the API as portable as possible it has been programmed using Java (*Java Technology*). Java can be run on common operating systems such as Windows, Mac OS and Unix/Linux.

To allow algorithms to be simply created the OOPic API handles all serial communication with the OOPic via the serial control protocol. This means that users are simply provided with objects from the API which allows easy access to the devices connected to the OOPic. Users simply need to create instances of the objects/devices that they which to use and set the properties of the object such as the pins which the device is connected to.

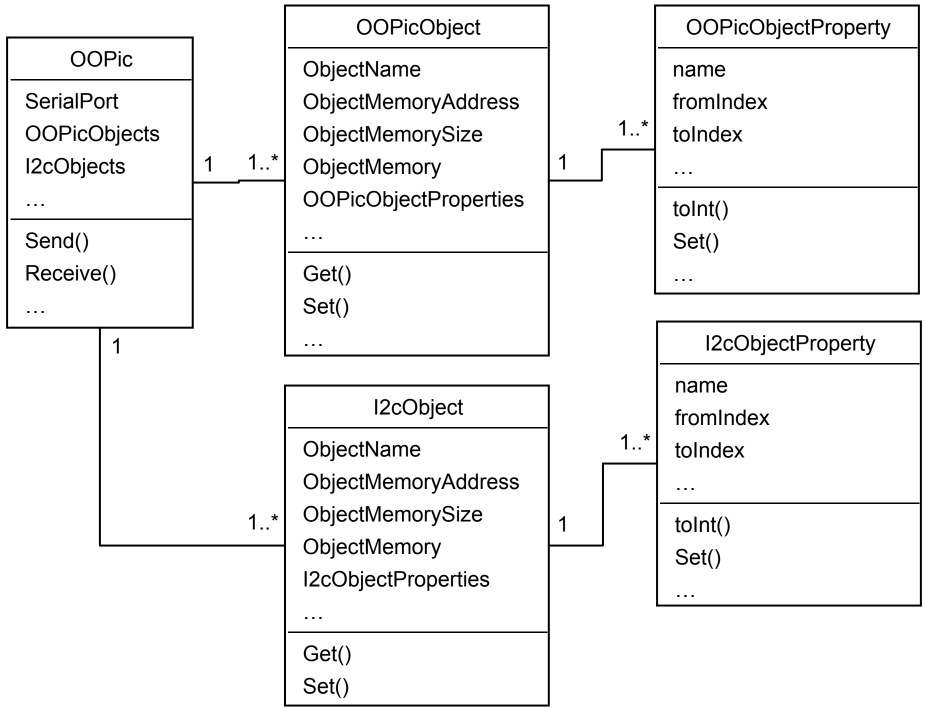


Figure 3.6 Simplified diagram of OOPic API

Figure 3.6 shows the basic structure of the OOPic API. Users simply create an OOPic instance, which is related to a particular serial port, then add either OOPic or I2C objects to the OOPic instance. The object properties are pre-defined and allow the setting of device properties, such as motor speed.

3.2.4.3 Implementation

The implementation of the OOPic API required the use of several procedures and also spurred the development of enhancements to the API.

In order to program the memory locations of device properties into the API a discovery process had to be used. The memory locations for properties of devices on the OOPic are not all provided in documentation, thus in order to determine the location of the properties in memory a discovery process was used. This involved programming the OOPic to modify a single device property and taking a snapshot of the devices memory before and after the change. The differences in the two snapshots then allowed the property's memory locations to be identified.

On examination of the OOPic programming it was discovered that objects are placed in memory in the order that they are created in the program loaded onto the OOPic. From this observation it was possible to create an automated code generator and also automatically determine the memory location of objects on the OOPic within the API. This provides two main advantages to users. Firstly, they do not need to know how to program the OOPic microcontroller, they can simply load on the automatically generated program. Secondly, they do not need to calculate the location of objects or object properties in the memory of the OOPic controller.

3.2.4.4 Documentation

Documentation is an important aspect of the API as it allows users to quickly and easily learn how to program it. In order to create a comprehensive set of documentation the automated Java documentation tool JavaDoc was used. This ensured that all aspects of the API were documented and also allowed additional information to be added to the documentation. Another advantage provided by JavaDoc is that as the API is extended the documentation can be easily updated at the same time. The documentation for the OOPic API is included in Appendix F.

3.2.5 Flexible Robot Platform

The modules, connection mechanisms, and software that have been brought together to form the Flexible Robot Platform (Figure 3.7), result in an easily configured and programmed robot platform. The platform also has ample space and mechanisms to allow the platform to be extended for use in a wide variety of applications. The total cost of the platform not including the laptop was around one thousand dollars.

Figure 3.7 The Flexible Robot Platform

Chapter 4 Testing and Analysis

In order to comprehensively test the Flexible Robot Platform, a number of tests and comparisons have been performed. This includes a comparison of the FRP with existing commercial platforms. Two quantitative navigational tests and a comparison of the motion capabilities of the FRP to investigate its usefulness in testing non-terrestrial mobile robot algorithms.

4.1 Flexibility Feature Comparison

In order to determine the flexibility of the Flexible Robot Platform its features have been compared against the flexibility factors identified in Figure 3.1 and the commercial mobile robot platforms introduced in 1.3. This allows a measure of how much flexibility was achieved in the creation of the FRP. The commercial platforms have been used for comparison as their specifications were more readily available and current.

4.1.1 Procedure

To allow a comparison between the platforms it is necessary to define each of flexibility factors, in order to determine whether each particular platform possesses the characteristic. Comparison against a defined set of factors also ensures that bias is not given to a particular platform. The definitions for each factor are defined in Table 2.

Factor/Characteristic	**Defined As**
Ability to carry additional processing	A laptop or small form factor PC can be added to its configuration.
Ability to interface third party components	Has available digital and analogue I/O lines, which can be accessed using its API
Laboratory or work bench operation	Can be operated within a laboratory or on a work bench
Large Payload Capacity	Can carry a maximum payload of over 10kg
Low Cost	Total cost of the working platform is under two thousand dollars.
Re-configurable chassis	Configuration of chassis can be changed without modification of the platform’s original design
Re-positionable sensors	Sensors on the platform can be easily moved to new positions on the chassis
Well developed and documented API	API provides access to all platforms features and is documented well enough to create new algorithms without extra help

Table 2 Flexibility factor definitions

4.1.2 Results

The results for the comparison are displayed in Table 3. If a platform fulfilled the definition for a factor it has been marked capable. Also, if a platform partially filled the definition it has been marked as limited. For platforms which require a laptop to complete the platform an additional one thousand dollars (the price of a low cost laptop) has been added to the platform's cost.

	ActivMedia Robotics Pioneer 1	Evolution Robotics ER-1 Personal Robot System	K-Team Khepera II	Lego Mindstorms	Nomad Super Scout II	Micro Robot "Alice"	Flexible Robot Platform
Ability to carry additional processing	■	■			■		■
Ability to interface third party components	□	■	□		□		■
Laboratory or work bench operation	■	■	■	■	■	■	■
Large payload capacity	■	□			■		■
Low cost		■		■		■	■
Re-configurable Chassis		■		■			
Re-positionable sensors		■		■			■
Well developed and documented API	■	□	■	■	■	■	■

■ Capable □ Limited

Table 3 Terrestrial mobile robots and their flexibility characteristics

4.1.3 Analysis

The comparison in Table 3 shows the Flexible Robot Platform has successfully implemented all of the identified characteristics of a flexible robot platform except the ability to re-configure its chassis. This is the result of a trade off where the benefits from a re-configurable chassis are not seen to outweigh the penalty to the Flexible robot platform's cost and payload capacity. This aside, the Flexible Robot Platform still possess more flexibility characteristics than any of the other mobile robot platforms examined.

4.2 Straight Line Test

In order to perform quantitative testing of the Flexible Robot Platform a series of three straight line navigation experiments developed and performed by Ibeanusi and others (1999) were performed using the Flexible Robot Platform. Ibeanusi and others performed the experiments using a Pioneer 1 mobile robot and have published their results, allowing a comparison of the performance between the Pioneer 1 and the FRP in this basic navigational task. However, the experiments also allow a test of the flexibility of the FRP as the FRP must adopt a sensor configuration and control algorithm to match that of the Pioneer 1 mobile robot. As the FRP is adopting a matching configuration it is expected that the FRP will perform similarly to the Pioneer 1 platform.

4.2.1 Procedure

The three experiments involve programming a mobile robot platform to travel in a straight line for 4877mm (16ft) using three different navigation methods. The first experiment (A) involves using a 'move' function to simply move the robot the specified distance. The second experiment (B) sets the robot's wheels to a common speed and monitors the cumulative distance travelled. The third experiment (C) steers the robot left or right depending on sonar information.

The Pioneer 1 robot is programmed using Saphira software in a C like language. However, this software is specific to the Pioneer robot platform and could not be used with the FRP. In order to implement the experiments a set of methods were created as an additional layer to the OOPic API developed for the FRP. The methods perform the following actions and their code can be found on the CD in Appendix E.

> ***setSpeed()***
>
> This method sets the speed of the FRP motors based on the desired speed for each of the FRP wheels. Each motor has a speed setting based on an 8-bit register (256 speed settings) that ranges from full reverse to full forward. Each of these speeds was calibrated by running the motor at each speed in the register for a few seconds. The calibrated speeds for each of the register

settings are stored in a lookup table, so that when a speed is selected the closest matching register setting can be selected and set.

readSonar()

This method operates the sonar sensor on the FRP and then returns a distance in millimetres. The distance is based on the amount of time taken for the sonar ‘ping’ to be echoed back to the sensor. If no echo is received then the function returns a distance of 0.

updatePosition()

This method reads the incremental change in wheel rotation from the FRP encoders and then calculates the position of the FRP using the well known equations for odometry (Klarer 1988, pp. 16-8), a convenient rewrite of these equations is also provided by (Borenstein, J, Everett & Feng 1996, p. 20).

move()

This method uses the *updatePosition()* and *setSpeed()* methods and attempts to steer the FRP in a straight line for the specified distance at the specified speed.

The creation of these methods allows the experiment programs to be written in a manner similar to that of the programs written by Ibeanusi and others. Before performing the experiments, the FRP was calibrated by instructing the platform to move 2 meters with the *move()* command. The actual distance travelled was recorded over 5 runs and averaged (a), then divided by the distance (d) and multiplied by the wheel diameter (Wd_a) to produce a new calibrated wheel diameter (Wd_b) (See Equation 1).

$$Wd_b = \left(\frac{a}{d}\right)Wd_a$$

Equation 1 FRP calibration

Each of the experiment algorithms are expressed in Figure 4.1 as pseudo-code, the actual Java code used by the FRP platform can be viewed on the CD in Appendix E.

Experiment A

```
move(4877);
```

Experiment B

```
while(robot.dist < 4877) {     // while robot has not travelled 4877mm
      setSpeed(200,200);       // set robot speed to 200mm/s
}
setSpeed(0);                   // set robot speed to 0 (stop)
```

Experiment C

```
keep_away = avg(5_sonar_reads);
while (robot.dist < 4877){     // while robot has not travelled 4877mm
      avg = avg(last_5_sonar_reads);
      If (avg < keep_away-50) {
            setSpeed(200,230);       // slowly turn left
      } else if ( avg > keep_away+50){
            setSpeed(230,200);       // slowly turn right
      }else {
            setSpeed(200,200);
      }
}
setSpeed(0);                   // set robot speed to 0 (stop)
```

Figure 4.1 Straight line experiment pseudo-code

4.2.1.1 Experiment A

Experiment A utilizes the *move()* method created as an additional layer to the FRP OOPic API for the straight line tests. It should be noted that although this algorithm is assumed to be similar to the code in Saphira software used by the Pioneer 1, this could not be confirmed as the code was not available for comparison.

4.2.1.2 Experiment B

Experiment B sets both wheels of the FRP to 200mm/s and then actively alters the speed of each motor to maintain this speed as closely as possible. It is assumed that the Pioneer 1 Saphira Software also functions in a similar manner. When 4877mm has been reached the wheels are stopped.

4.2.1.3 Experiment C

Experiment C sets both wheels of the FRP to 200mm/s and then, depending on the sonar sensor, moves one wheel faster at 230mm/s if the robot is 50mm further away or 50mm too close to the wall with respect to the computed *keep_away* value (See Figure 4.1). To implement this experiment on the FRP, two changes from the implementation by Ibeanusi and others were made. Firstly, Ibeanusi and others implemented a weighted average for the robot sonar readings. However, as the weighting system used is unknown a weighted average was not used in the FRP implementation. Secondly, rotation of the robot if it exceeds 50mm from the *keep_away* value is facilitated by setting the speed of one of the wheels to 230mm/s, which may have resulted in a change in rotation slightly more than 2 degrees per second as implemented by Ibeanusi and others.

1. Robot aligned at start point 508mm from wall
2. Experiment program run
3. Experiment run to completion or stopped if robot collided with wall
4. Distance from start point recorded
5. Distance from wall recorded

Figure 4.2 Straight line experiment procedure

Each experiment was run 24 times, using the procedure in Figure 4.2. The experiments were performed against a straight length of wall 5 meters in length and on a dense carpet surface. For convenience the starting position for the robot was marked on the floor using electrical tape and positioned at this point at the start of each test. The centre line of the platform was also marked for measurement purposes. Figure 4.3 shows how the experiment area was setup.

After performing the experiments it was noted that Ibeanusi and others(1999) had in fact positioned the Pioneer 1 platform's centreline 508mm from the wall rather than the entire platform. This has produced a disparity between the results from the FRP and Pioneer 1 platforms. In order to eliminate this problem the Y travel values for FRP have been reduced by half the width of the platform (177.5mm), so that they are in the same domain as the Pioneer 1 results. This will not affect the performance of the FRP platform in any way in experiments A or B, as the wall is simply used as a common point of measure, however, it will have an affect on experiment C, where the wall is used for sonar measurement by the robot platform. This factor is discussed in more detail in section 4.2.3.

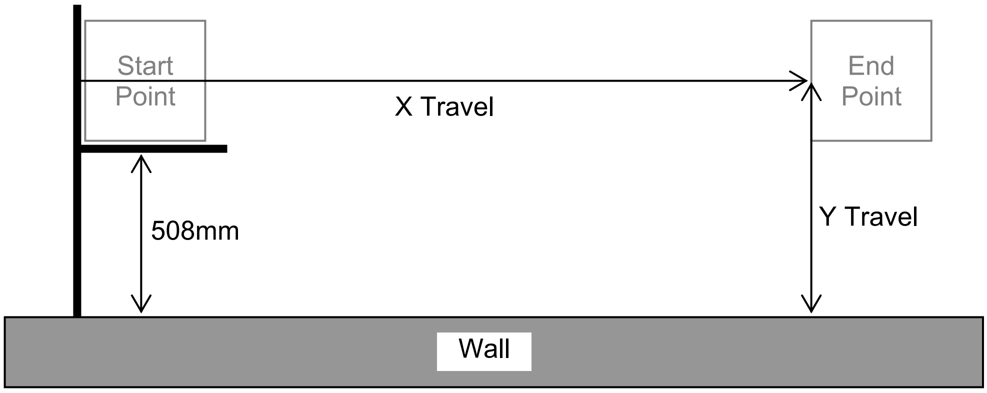


Figure 4.3 Experiment area setup

4.2.2 Results Run 1

The data recorded from running each of the experiments is located in Appendix A. For each of the straight line experiments run, the mean, standard deviation and t statistics were computed as in the paper by Ibeanusi and others (1999, p. 3), using equations 2, 3 and 4.

$$x' = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equation 2 Mean distance

$$s = \sqrt{\frac{\sum (x_i - x')^2}{n-1}}$$

Equation 3 Standard deviation

$$t = \frac{x' - \mu}{s / \sqrt{n}}$$

Equation 4 Students t statistic

The mean provides an indication of how far the robot is travelling in each of the three experiments. The standard deviation shows the amount of spread between each of the travel values and the mean. The t statistic provides an indication of how well the sampling distribution represents the real distribution for each experiment with respect to the target travel for both X and Y. According to Ibeanusi and others (1999) a t statistic value between -3.0 and +3.0 can be attributed to chance, whereas a value out of this range indicates that some kind of systematic error is occurring in the robot's control.

The following results have Y travel values adjusted for comparison with the Pioneer 1 platform, however the recorded data for each of the experiments is included in Appendix A.

Statistics in mm	Experiment A		Experiment B		Experiment C	
	X Travel	Y Travel	X Travel	Y Travel	X Travel	Y Travel
Target μ	4877	508	4877	508	4877	508
Mean Travel x'	4742.8	395.9	4848	399	4843	535
Standard Deviation s	10.45	78.39	8	82	16.5	241
Student's t Statistic t	-65.64	-7.03	-17.76	-6.51	-10.09	0.55

Table 4 Straight line test statistics with Pioneer 1 (Ibeanusi et al 1999)

Statistics in mm	Experiment A		Experiment B		Experiment C	
	X Travel	Y Travel	X Travel	Y Travel	X Travel	Y Travel
Target μ	4877	508	4877	508	4877	508
Mean Travel x'	5193.25	354.2917	4812.458	465.5	4303.667	558.0833
Standard Deviation s	786.2355	307.7146	179.4164	256.9688	57.35827	102.7212
Student's t Statistic t	1.970532	-2.44712	-1.76232	-0.81024	-55.8215	2.63042

Table 5 Straight line test statistics with FRP run 1

The statistical values for each of the experiments calculated for both X and Y travel using the FRP are presented in Table 5 and the values for the Pioneer 1 platform experiments as performed by Ibeanusi and others (1999) in Table 4. On examination of the mean and t statistic values for the FRP it appears that very little systematic error is occurring. However, the standard deviation values for the experiment are far greater than would be expected, especially when compared with the results obtained by Ibeanusi and others (1999, pp. 4-5) using a Pioneer 1 robot platform. As the FRP was expected to perform similarly, with differences from wheel slippage and internal errors, there is indication of an error in a part of the FRP implementation. The problem is clearly shown in Figure 4.4.

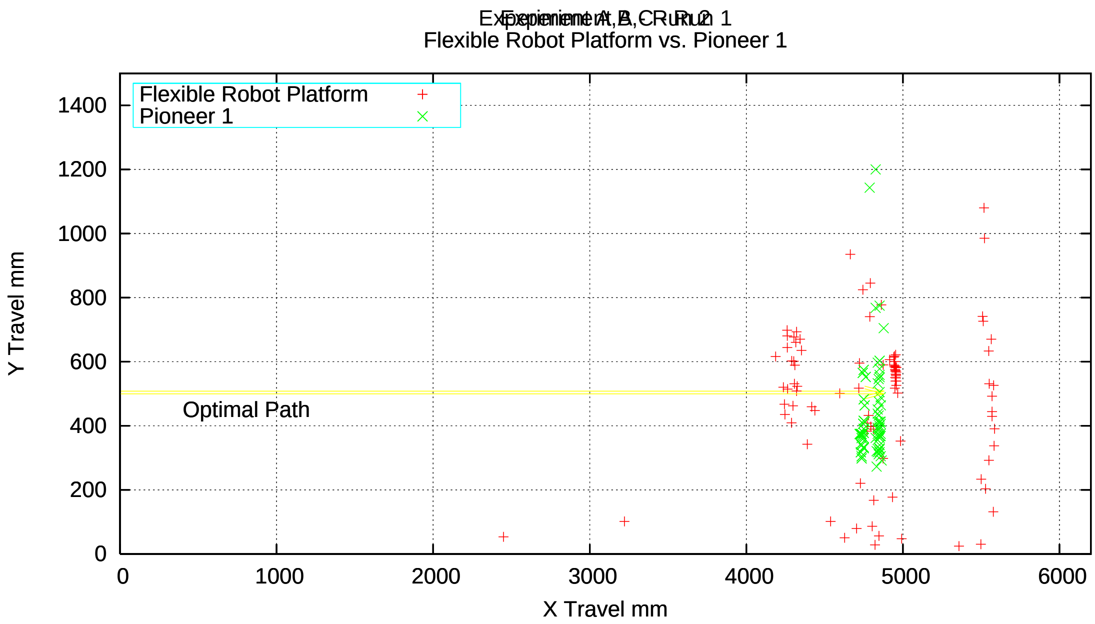


Figure 4.4 Straight line test, experiments A, B & C

The indication of an error by the results prompted an analysis of the FRP code to determine what might be causing the poor results of the FRP. The analysis of code exposed a problem in the code which retrieves encoder data. The code was using a process of pause-read-clear-play to obtain encoder readings from an encoder register on the OOPic microcontroller. However, as the wheels would continue to turn whilst performing this process, the rotational information generated during the reading process was being lost. The algorithm was fixed by modifying the algorithm to only read the encoder register and determine the amount of rotation from the difference compared to the last encoder read. This ensures that no encoder information is lost.

Subsequent to fixing the error in the algorithm for the experiments, each of the experiments were re-run in order to produce a more accurate comparison between the FRP and Pioneer 1 robot platforms.

4.2.3 Results Run 2

Before running the experiments for the second time the FRP platform was re-calibrated as the previous calibration would have been inaccurate for the platform with the new encoder reading method. The FRP was calibrated again using Equation 1, however this time a distance of 4 meters was used 10 times in order to produce a more accurate calibration.

Statistics in mm	Experiment A		Experiment B		Experiment C	
	X Travel	Y Travel	X Travel	Y Travel	X Travel	Y Travel
Target μ	4877	508	4877	508	4877	508
Mean Travel x'	4952.458	569.0833	4951.792	548.25	4970.542	533.5833
Standard Deviation s	5.897525	30.63376	7.002976	115.7472	10.20221	48.11121
Student's t Statistic t	62.68203	9.768502	52.32102	1.703574	44.91758	2.605053

Table 6 Straight line test statistics with FRP run 2

The data recorded for the second run of the straight line test can also be found in Appendix A. The statistics generated from the results gathered for the second run of the straight line test on the FRP (Table 6) are greatly improved compared to the first run and, as was originally expected, are similar to the statistical results (Table 4) from Ibeanusi and others (1999) with the Pioneer 1 platform. In all of the experiments the FRP moves further than the target distance of 4877mm with a small amount of deviation. This would indicate that there is some kind of systematic error such as an inaccuracy in the FRP calibration or controlling program causing it to travel further than intended. The Y travel statistics for the experiments however are dissimilar and will be examined individually.

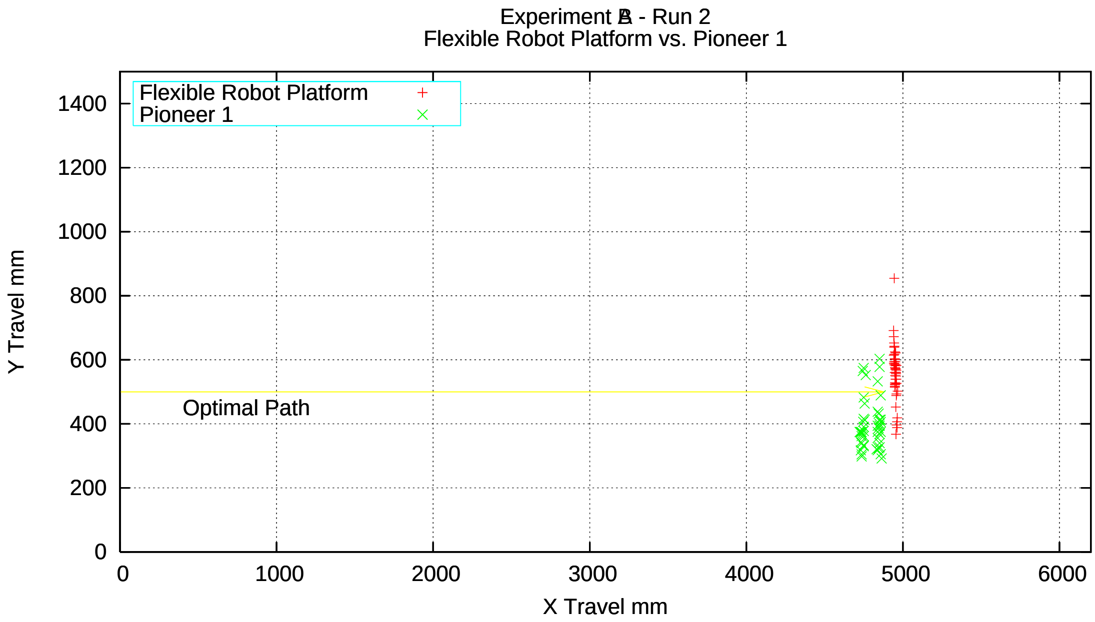


Figure 4.5 Straight Line Test: Experiment A Run 2

Experiment A shows that the FRP manages to more consistently move in a straight line than the Pioneer 1 platform, achieving a tighter clustering of end points. This is reflected by the FRP's Y travel standard deviation of 30.63 as opposed to 78.39 achieved by the Pioneer 1. However, the FRP appears to veer consistently to the left and, as observed by Ibeanusi and others, (1999, p. 4) the Pioneer 1 platform appears to drift to the right (towards the wall). The more consistent results achieved by the FRP may be the result of two main factors. Firstly, the control algorithm may be superior to that of the Pioneer 1 in this circumstance. The second and more likely reason for the improvement is the difference in encoder resolution of each of the platforms. The Pioneer 1 having a resolution of 100 ticks per revolution (*Pioneer Mobile Robots: Operation Manual* 1998) and the FRP 500 ticks per revolution, providing the FRP with greater resolution and thus the ability to more accurately control its motion.

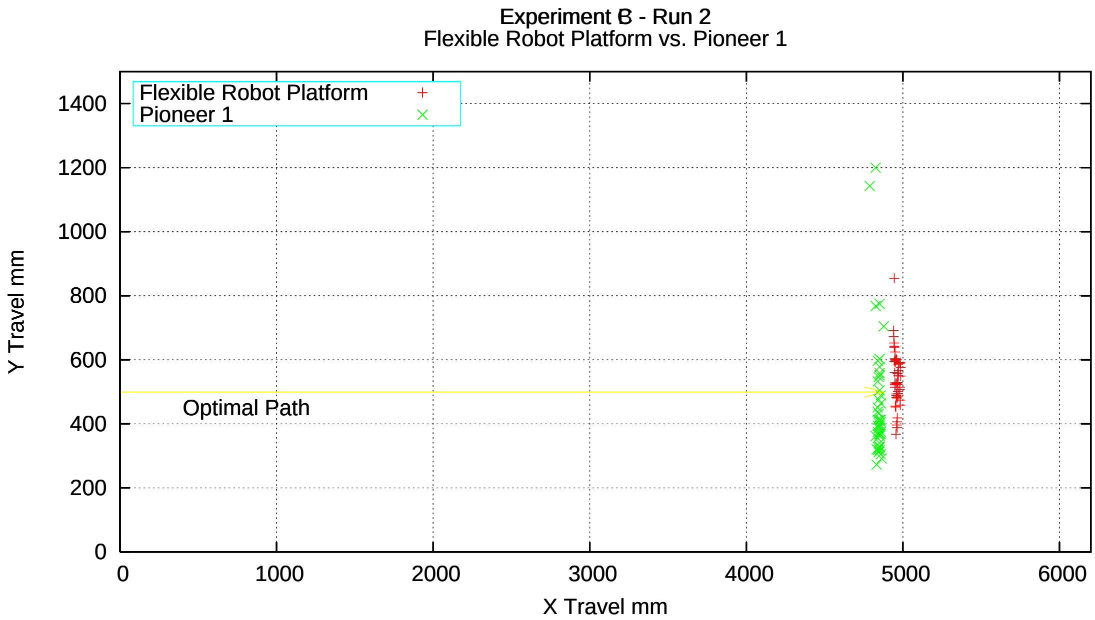


Figure 4.6 Straight Line Test: Experiment B Run 2

Experiment B produced similar results for both platforms. The FRP has a higher standard deviation for Y travel. However, on examination of the graphed results in Figure 4.6 this would appear to be due to a noisy result. The veering of the robot platforms to the right or left appears to be occurring as in experiment A to a lesser extent.

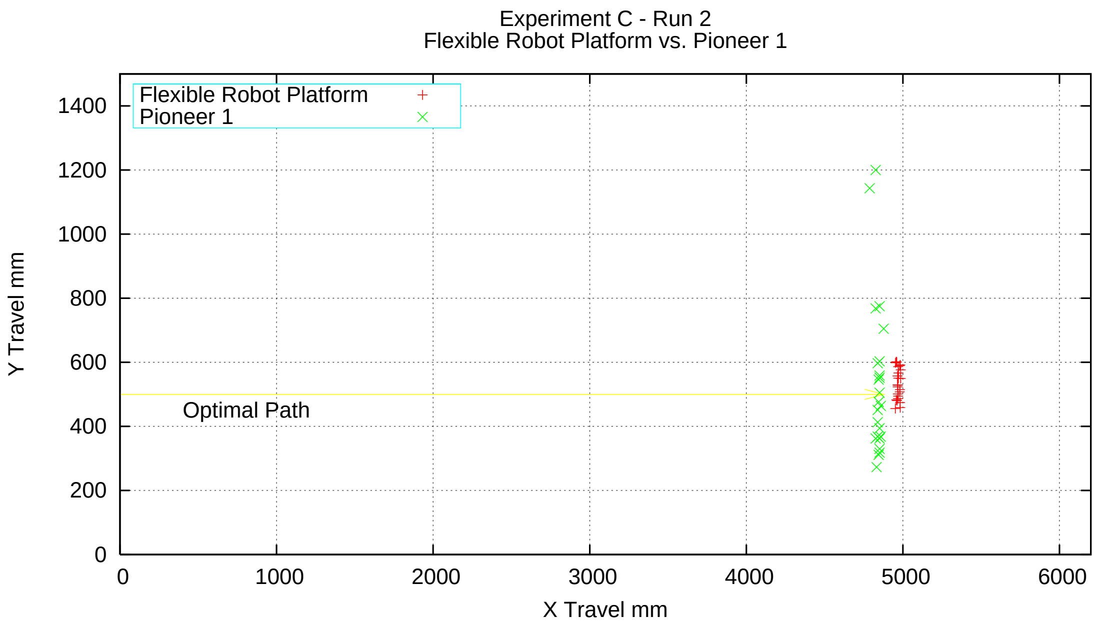


Figure 4.7 Straight Line Test: Experiment C Run 2

Although the results for the second run of experiment C indicate that the FRP performs better for Y travel, the results are inconclusive due to a number of factors relating to the use of sonar. Firstly, the platforms were situated at different distances from the wall, the FRP being placed further away as the platform was placed 508mm form the wall rather than its centre line. This should reduce the performance of the FRP, however, as the accuracy of sonar sensors decreases with distance due to their cone shaped sensitivity (Nehmzow 2003, p. 29). The second factor affecting the experiment is the use of different sonar sensors on the FRP and Pioneer 1 platform. The FRP uses a SRF04 sonar sensor which has a range of 3cm to 3m where as the Pioneer 1 has a Polaroid type sensor which has a range of 15cm to 5m. This provides the FRP with a higher resolution over the shorter distance. In addition the FRP may be sampling its sonar sensor at a higher rate resulting in a more current reading of the distance to the wall and a faster response to correct Y travel error.

4.2.4 Analysis

The second run of the experiments with the FRP show that the platform can perform as well as, if not better than, the Pioneer 1 platform. The FRP achieved a lower standard deviation than the Pioneer 1 platform on all measurements except the Y travel in experiment B. Also, the test shows that the FRP platform has been successful in its non performance based objectives. The FRP has been able to successfully perform an experiment designed for another mobile robot platform (the Pioneer 1). In addition, the FRP allowed the identification of an error in the platform's control algorithm through running the experiments. This is significant as the initial algorithm was thought to be correct and so the FRP has been an important tool in improving the algorithm.

4.3 UMBmark test

In order to provide further quantitative measures of the Flexible Robot Platform the UMBmark (University of Michigan Benchmark test) test developed by Borenstein and others at the University of Michigan (1995) was run using the Flexible Robot Platform. The test provides a measure of the odometric accuracy of a platform for systematic errors.

4.3.1 Procedure

The UMBmark test involves running the robot platform along a pre-programmed 4x4m square path in both the clockwise and counter clockwise direction (see Figure 4.9). The robot is run in a clockwise (CW) and counter clockwise (CCW) direction to eliminate the problem where odometry errors are concealed due to the fact they compensate for each other (Borenstein, Johann & Feng 1995, p. 5). Running the robot in the CCW direction should identify these errors. The position of the robot is measured against a reference wall both before and after the robot has performed the square path and compared to the position calculated by the robot's internal odometry. In order to run the experiment using the FRP the methods created for the straight line tests (section 4.2) were used with the addition of a new method which turns the FRP a specified number of degrees (code is included on the CD in Appendix E).

1. At the beginning of the run, measure the absolute position of the vehicle and initialize to that position the starting point of the vehicle's odometry.
2. Run the vehicle through a 4x4m square path in CW direction ensuring to:
   - Stop after each 4m leg
   - Make a total of four 90° turns on the spot
   - Run the vehicle slowly to avoid slippage
3. Upon returning to start area, measure the absolute position of the vehicle.
4. Compare the absolute position to the robot's calculated position, based on odometry using Equation 5.
5. Repeat steps 1-4 for four more times (i.e., a total of five runs).
6. Repeat steps 1-5 in the CCW direction.
7. Use equations 6, 7 and 8 to express the experimental results quantitatively as the measure of odometric accuracy for systematic errors $E_{\max.syst}$.

Figure 4.8 Summary of the UMBmark procedure (adapted from (Borenstein, Johann & Feng 1995, p. 7))

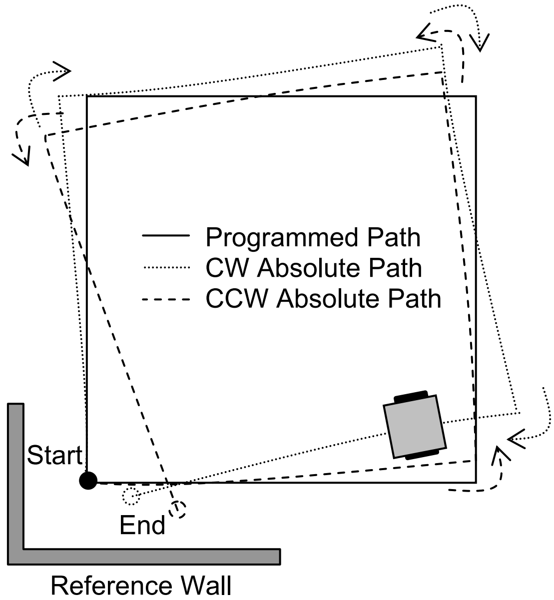


Figure 4.9 UMBmark square path and associated error (adapted from (Borenstein, Johann & Feng 1995, p. 5))

Once the experiment has been run 5 times in both directions (10 runs total) the UMBmark score can be determined for the robot platform using the following equations created by Borenstein and others (1995).

For each of the runs a set of return position errors can be computed from the calculated and absolute positions.

$$\in x = x_{abs} - x_{calc}$$

$$\in y = y_{abs} - y_{calc}$$

where

$\in x, \in y$ - Position errors due to odometry

x_{abs}, y_{abs} - Absolute position of the robot

x_{calc}, y_{calc} - Position of the robot computed form odometry

Equation 5 UMBmark: Return position error

From the return position errors (Equation 5) two 'centres of gravity' (average of the runs in each direction) can be computed. Computing the 'centres of gravity' reduces the effect of non-systematic errors (Borenstein, Johann & Feng 1995, p. 5).

$$x_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} \in x_{i,cw/ccw}$$

$$y_{c.g.,cw/ccw} = \frac{1}{n}\sum_{i=1}^{n} \in y_{i,cw/ccw}$$

Where $n = 5$ is the number of runs in each direction.

Equation 6 UMBmark: Centres of gravity

The absolute offset of the two centres of gravity (Equation 6) from the origin can then be found. These two values are denoted $r_{c.g.,cw}$ and $r_{c.g.,ccw}$.

$$r_{c.g.,cw} = \sqrt{(x_{c.g.,cw})^2 (y_{c.g.,cw})^2}$$

And

$$r_{c.g.,ccw} = \sqrt{(x_{c.g.,ccw})^2 (y_{c.g.,ccw})^2}$$

Equation 7 UMBmark: Absolute offsets

The larger of the two values ($r_{c.g.,cw}$ and $r_{c.g.,ccw}$) is then selected as the measure of odometric accuracy for systematic errors.

$$E_{\max.syst} = \max(r_{c.g.,cw}; r_{c.g.,ccw})$$

Equation 8 UMBmark: Measure of odometric accuracy for systematic errors

$E_{\max.syst}$ forms the single numeric value or UMBmark 'score' for evaluation of the platform, and indicates the maximum odometry error of a platform. A low maximum systematic error indicates that a platform is accurately computing its position allowing it to move with greater accuracy. A summary of the UMBmark Procedure adapted from (Borenstein, Johann & Feng 1995) is provided in Figure 4.8.

Before running the UMBmark procedure using the FRP it was calibrated using Equation 1 over a distance of 4 meters with 10 trials. The FRP was also run at 50mm/s during the calibration as this speed was used for running the UMBmark test. Running the platform slowly is suggested by (Borenstein, Johann & Feng 1995, p. 7) to avoid slippage. Also, testing with the FRP for UMBmark had to be performed on a surface of dense carpet tiles as a 4x4m area of smooth concrete was unavailable. Normally this would have a detrimental effect on performance, however the computing of the ‘centres of gravity’ (Equation 6) minimizes the effect this will have on results.

4.3.2 Results

The results running the UMBmark benchmark test on the FRP are displayed in Table 7, along with the results obtained by (Borenstein, Johann & Feng 1995) with a TRC LabMate platform under various conditions. The TRC LabMate is a suitable platform for comparison as it has a similar wheel base (340mm) as the FRP (335mm).

Platform Name	Modification	Calibration	$E_{\max.syst}$
TRC LabMate	None	none	310
TRC LabMate	3 loops of masking tape on right wheel	none	423
TRC LabMate	None	UMBmark	26
TRC LabMate	3 loops of masking tape on right wheel	UMBmark	20
Flexible Robot Platform	**None**	**basic (Equation 1)**	**612**

Table 7 UMBmark results from Flexible Robot Platform and TRC LabMate (Borenstein, Johann & Feng 1995)

Also, the return position error for the FRP has been plotted in Figure 4.10, showing the error from the target (0,0) and the ‘centres of gravity’ for CW and CCW runs. The recorded data for the UMBmark tests is included in Appendix B.

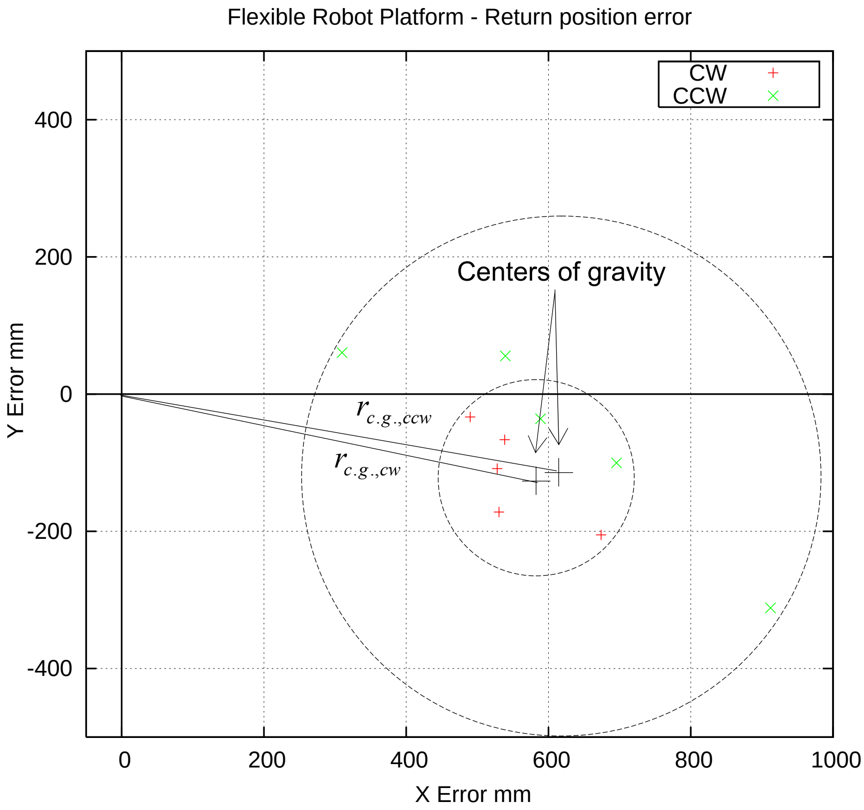


Figure 4.10 Return position error for Flexible Robot Platform

4.3.3 Analysis

The FRP UMBmark results show that it did not perform as well as the LabMate platform used by (Borenstein, Johann & Feng 1995) even with a basic calibration. This can be attributed to two main factors, the basic nature of the calibration method and the physical traits of the FRP's drive wheels.

The calibration method used with the FRP (Equation 1) is very simple, only taking into account the travel of the robot on a single axis. This method cannot identify or correct factors such as uneven wheel diameters (which cause a robot to arc) or uncertainty about the effective wheel base, which are cited by Borenstein and others (1996) as the two dominant error sources in differential-drive mobile robots. As the basic calibration method modifies the wheel diameter of the mobile robot it may in fact be amplifying the effect of the previously stated factors.

The FRP's drive wheels do not have a single point of contact, making it difficult to determine the effective wheel base. This affects the ability of the robot to effectively calculate its position. In addition, the drive wheels have a number of imperfections on their tread which may result in uneven wheel diameters.

Although the result of the UMBmark test reflects negatively on the calibration method used on the FRP, the results are a positive outcome when considering the flexibility of the FRP. Firstly, the FRP was able to successfully run the UMBmark test, confirming that the FRP can indeed implement algorithms. This has also been shown in section 4.2 with the straight line test. Secondly, that FRP has provided a tool for extracting meaningful results on the performance of implemented algorithms (in this case the calibration method). It is reasonable to assume that the FRP could be made to perform better in the UMBmark test by following the compensatory calibration technique developed by Borenstein and others (1996, pp. 13-9), however the implications to the flexibility of the FRP remain the same. Whether an algorithm is calibrated to be as accurate as possible is a factor which is dependent on the nature of the algorithm being tested and so calibration is a testing condition that needs to be considered. For example, if a new algorithm is being tested, which attempts to accurately calculate position without knowledge of the drive system (perhaps vision based), calibration would not be required. Additionally, calibrating the FRP for high accuracy may limit the exposure of an algorithm to real world anomalies.

4.4 Mobile Platform Motion Comparison

Having established that the FRP can indeed implement algorithms for terrestrial mobile robots, the question was raised whether the FRP could be used to test algorithms for other kinds of mobile robot platforms such as airborne or aquatic platforms. This would be advantageous, as aquatic and airborne platforms are difficult to test due to the fact that they require special testing environments and are inaccessible during tests (Kuroda, Aramaki & Ura 1996, p. 365).

It is observed that the main difference between terrestrial, aquatic, airborne and space platforms are their motion capabilities. Thus the FRP should be capable of implementing algorithms for other types of platforms with which it shares motion capabilities. In order to determine which types of platforms the FRP can implement algorithms for, an analysis of the motion capabilities for a number of real terrestrial, aquatic and airborne platforms has been performed.

4.4.1 Classification technique

The arbitrary motion of objects (mobile robots) whether in air, space or water requires six degrees of freedom (Eaton et al. 2002, p. 3). However mobile robots usually possess 2 or more degrees of freedom. For each platform considered, the degrees of freedom it was capable of was recorded. This included the platform's ability to Roll, Pitch and Yaw and to move along each of these axes without first turning (see Figure 4.11). In order to discern a platform's motion capability the specifications, diagrams and photographs available for the platform were examined. The capabilities for each of the platforms were then summarized into generic motion capabilities for similarly designed mobile robots, in order to determine the FRP's capabilities for implementing other platform type algorithms in broad types rather than specific platforms.

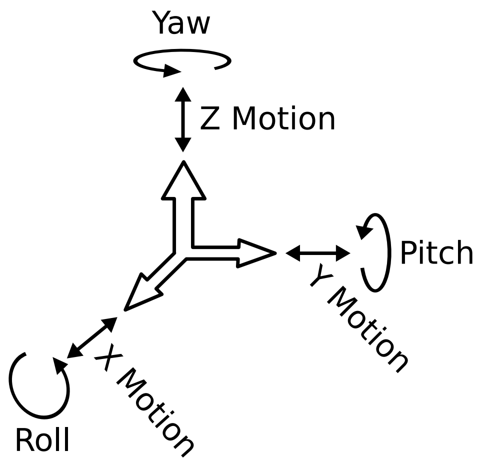


Figure 4.11 Mobile Robot Degrees of Freedom

4.4.2 Result

The summarized table of motion capabilities is presented in Table 1 with the table of all the platforms considered included in Appendix C. The motion capabilities for a thrusters based aquatic platform in Table 8 assumes that the platform has thrusters for each degree of freedom, however not all the thrusters based platforms had thrusters for all degrees of freedom.

Platform Type	Turning Circle			X Motion		Y Motion		Z Motion	
	Roll	Pitch	Yaw	Forward	Reverse	Left	Right	Up	Down
Terrestrial									
Flexible Robot Platform	**N**	**N**	**0**	**Y**	**Y**	**N**	**N**	**N**	**N**
Differential Drive	N	N	0	Y	Y	N	N	N	N
Ackerman (Car)	N	N	>0	Y	Y	N	N	N	N
Omni-directional	N	N	0	Y	Y	Y	Y	N	N
Aquatic									
Single Propeller Underwater Vehicle	>0	>0	>0	Y	Y	N	N	N	N
Thrusters Based Underwater Vehicle	0	0	0	Y	Y	Y	Y	Y	Y
Surface Craft (Boat)	N	N	>0	Y	Y	N	N	N	N
Airborne									
Single Propeller Aerial Vehicle (Plane)	>0	>0	>0	Y	N	N	N	N	N
Helicopter	>0	>0	0	Y	Y	Y	Y	Y	Y

Table 8 Mobile Robot Motion Capabilities

4.4.3 Analysis

In establishing the platforms for which the FRP would be able to implement algorithms the data in Table 8 was examined. The data indicates that there are two other platform types for which the FRP shares motion capabilities, terrestrial Ackerman or car type platforms and aquatic surface craft or boats. This is a relatively small set of platform types for which the FRP can be used to implement algorithms. The main reason for the FRP not being able to share motion capabilities with most platform types is the fact that it cannot pitch or roll. This is due to its inability (like all terrestrial platforms) for arbitrary Z axis motion.

With the addition of the ability to roll and pitch (although not possible) the FRP would also be able to match the motion capabilities of single propeller aerial and underwater vehicles. One solution to this problem is to simulate or 'dummy' the Z axis information in the controlling programs of the robot, in a hybrid approach. Whilst the simulation of some motion aspects would introduce the problems of simulation, their effect would be diminished. Thus testing an algorithm in a hybrid approach would still be preferable to an entirely simulated test. Kuroda and others have found a hybrid approach to still be beneficial in the development of algorithms for an autonomous underwater vehicle (Kuroda, Aramaki & Ura 1996).

Chapter 5 Conclusion

The implementation and testing of the Flexible Robot Platform has provided the necessary evidence to evaluate the initial hypothesis:

That it is possible to create a mobile robot platform, with the flexibility to test a variety of algorithms.

The results gathered reflect positively on the hypothesis. During the construction of the Flexible Robot Platform factors which determine the flexibility of a mobile robot platform were identified. These factors were able to be successfully integrated into the FRP. During the construction of the FRP various methods of achieving flexibility were also produced. The use of off-the-shelf components supported the claims in previous research of allowing greater flexibility (Salemi et al. 2005).

Comparison of the Flexible Robot Platform with existing mobile robot platforms showed that it achieved a high level of flexibility. With room for improvement in the area of chassis configuration, this may form the basis for future work.

The straight line test showed that the FRP achieved a number of critical goals. Firstly, the FRP was able to perform as well as if not better than an existing mobile robot platform. Secondly the sensors of the FRP were successfully put into the desired configuration (in this case mimicking the Pioneer 1 mobile robot). Thirdly and most importantly the use of the FRP allowed the identification and rectification of a problem with an algorithm, the outcome of which, provided a significant improvement in the algorithms performance.

The UMBmark test showed that the FRP can implement different algorithms with varying requirements. The test also showed that the FRP greatly exposes algorithms to the nuances of operating in a real world.

The analysis of the motion capabilities of different kinds of mobile robot platforms compared to the Flexible Robot Platform identified an approach to algorithm testing which merits further investigation. If successful this technique would allow research into difficult to test mobile robot platforms to proceed more easily.

In conclusion, the Flexible Robot Platform provides a valuable tool for research, allowing the testing of a variety of algorithms, through its flexible design. The platform shows promising results for the further development of flexible robot platforms. Use of the Flexible Robot Platform on research projects with yet unknown requirements will provide an unequivocal test of the platform's ability to function as a truly re-usable mobile robot platform.

Chapter 6 Future Work

The development of the Flexible Robot Platform has provided the opportunity for a great variety of research paths. The paths which relate to extending the flexibility of mobile robot platforms will be explored.

6.1 Re-configurable Chassis and Drive System

Due to cost and implementation factors, the FRP has an inflexible chassis and drive system. In future developed platforms or modifications to the FRP it would be advantageous to have a re-configurable chassis and drive system. This would allow better testing of algorithms which are affected by the shape of the mobile robot platform and the way that the platform propels itself in the environment. For example this would allow an autonomous underwater vehicle to be modelled more accurately by the flexible platform.

6.2 Higher Complexity Algorithms

The algorithms implemented on the FRP were only of moderate complexity. In order to further test the FRP flexibility of the platform it would be ideal to implement a high complexity algorithm. The testing of more algorithms with the FRP will further prove the case for the development of flexible mobile robot platforms.

6.3 Linking to Open Source Software

Open source projects may provide a way to further increase the flexibility of the FRP. Linking with open source projects such as The Open Robot Control Software (OROCOS) Project (*The OROCOS Project*) would provide a collaborative development environment for developers and users of the platform, with obvious benefits. The re-use and sharing of code between developers would prevent developers from writing separate algorithms to achieve the same task. If the code to implement algorithms is already available as open source, implementation times will be reduced allowing more time to be allocated to important research tasks such as more extensive testing.

References

ActivMedia Robotics, viewed 10 May 2005, <http://www.activrobots.com/>.

Agilent Technologies 2001, *Agilent Technologies*, viewed 15 June 2005, <http://www.home.agilent.com/>.

Bergbreiter, S & Pister, KSJ 2003, 'CotsBots: An Off-the-Shelf Platform for Distributed Robotics', paper presented to IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada.

Borenstein, J & Feng, L 1995, 'UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots', paper presented to SPIE Conference on Mobile Robots, Philadelphia.

---- 1996, 'Measurement and Correction of Systematic Odometry Errors in Mobile Robots', *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869-80.

Borenstein, J, Everett, HR & Feng, L 1996, *"Where am I?" Sensors and Methods for Mobile Robot Positioning*, The University of Michigan.

Bose, AK 2004, *Modular Omni Directional Drive for a Mobile Robot Platform*, viewed 30 October 2005, <http://www.intelligentcomponents.com/abheek/docs/fne11.pdf>.

Brooks, R 1992, *Artificial Life and Real Robots*, viewed 30 October 2005, <citeseer.ist.psu.edu/brooks92artificial.html>.

Caprari, G, Balmer, P, Piguet, R & Siegwart, R 1998, 'The Autonomous Micro Robot "Alice": a platform for scientific and commercial applications.' paper presented to Proceedings of the 1998 International Symposium on Micromechatronics and Human Science, 1998. MHS '98., Nagoya.

Challinger, J 2005, 'Efficient use of robots in the undergraduate curriculum', in *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, ACM Press, St. Louis, Missouri, USA, pp. 436-40.

Devantech Ltd 2003a, *SRF04 - Ultra-Sonic Ranger*, viewed 11 October 2005, <http://www.robot-electronics.co.uk/htm/srf04tech.htm>.

---- 2003b, *CMPS03 - Robot Compass Module*, viewed 11 October 2005, <http://www.robot-electronics.co.uk/htm/cmps3doc.shtml>.

Dozier, G 2001, 'Evolving robot behaviour via interactive evolutionary computation: from real-world to simulation', in *Proceedings of the 2001 ACM symposium on Applied computing*, ACM Press, Las Vegas, Nevada, United States, pp. 340-4.

Dudek, G & Jenkin, M 2000, *Computational Principles of Mobile Robotics*, Cambridge University Press.

Eaton, D, Bobbitt, M, Reid, M, Jarvis, G, Frame, S, Dowling, J, Bloechl, M, Mentzer, M & Manseur, R 2002, *The Nautilus-Sub Autonomous Underwater Vehicle*, viewed 29 October 2005, <http://www.auvsi.org/competitions/2002/papers/UWFlorida.pdf>.

Encyclopaedia Britannica 2005, *robot*, Encyclopaedia Britannica Premium Service, viewed 9 October 2005, <http://www.britannica.com/eb/article-9063935>.

ER1 Personal Robot System, 2005, Evolution Robotics, viewed 9 May 2005, <http://www.evolution.com/er1/>.

Fagin, B & Merkle, L 2003, 'Measuring the effectiveness of robots in teaching computer science', in *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, ACM Press, Reno, Nevada, USA, pp. 307-11.

Gerecke, U, Hohmann, P & Wagner, B 2003, 'Educational Robotics in a Systems Design Masters Program', paper presented to 3rd IEEE International Conference on Advanced Learning Technologies, Athens, Greece.

Ibeanusi, U, Lafond-Favieres, V, McLawhorn, L & Anderson, SD 1999, 'Understanding robot navigation and control', in *Proceedings of the 37th annual Southeast regional conference (CD-ROM)*, ACM Press, p. 42.

Java Technology, viewed 30 October 2004, <http://java.sun.com/>.

Klarer, PR 1988, *Simple 2D Navigation for Wheeled Vehicles*, SAND88-0540 UC-15, Sandia National Laboratories.

K-Team Corporation, 2005, K-Team Corporation, viewed 10 May 2005, <http://www.k-team.com/index2.html>.

Kuroda, Y, Aramaki, K & Ura, T 1996, 'AUV Test using Real/Virtual Synthetic World', paper presented to IEEE Conference on Autonomous Underwater Vehicle Technology, Monterey, CA, 3-6 June.

Lee, T-m, Nehmzow, U & Hubbold, R 1998, 'Mobile Robot Simulation by Means of Acquired Neural Network Models', paper presented to European Simulation Multiconference, Manchester.

Lego(tm) Mindstorms(tm), 2005, Lego, viewed 13 May 2005, <http://mindstorms.lego.com/eng/products/ris/risdetails.asp>.

Lima, PU & Custódio, LMM 2004, 'Artificial Intelligence and Systems Theory Applied to Cooperative Robots', *International Journal of Advanced Robotic Systems*, no. 3.

Loose, H, Boersch, I, Heinsohn, J & Mrkor, K-U 2004, 'RCUBE - A MULTIPURPOSE PLATFORM FOR MOBILE ROBOTS', paper presented to IFAC Symposium on Mechatronic Systems, Sydney, Australia.

Maxwell, BA & Meeden, LA 2000, 'Integrating robotics research with undergraduate education', *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]*, vol. 15, no. 6, pp. 22-7.

McComb, G 2001, *The Robot Builder's Bonanza*, 2nd edn, McGraw-Hill.

Michel, O 2004, 'WebotsTM: Professional Mobile Robot Simulation', *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 40-3.

Nehmzow, U 2003, *Mobile Robotics: A Practical Introduction*, Second edn, Springer-Verlag London Limited.

Ono, Y, Uchiyama, H & Potter, W 2004, 'A mobile robot for corridor navigation: a multi-agent approach', in *Proceedings of the 42nd annual Southeast regional conference*, ACM Press, Huntsville, Alabama, pp. 379-84.

The OROCOS Project, viewed 30 October 2005, <http://www.orocos.org/>.

Outfitting A Robot Laboratory, Bryn Mawer, viewed 10 May 2005, <http://mainline.brynmawr.edu/Robots/ResourceKit/Lab.html>.

Philips Electronics 2005, *Philips Semiconductors I²C-bus Information*, Koninklijke Philips Electronics N.V., viewed 13 May 2005, <http://www.semiconductors.philips.com/markets/mms/protocols/i2c/>.

Pioneer Mobile Robots: Operation Manual, 1998, ActivMedia Incorporated, viewed 10 October 2005, <http://www.ing.unibs.it/~cassinis/docs/documentation/Pioneer%201%20documentation/opmanv2.pdf>.

Salemi, B, Reis, J, Saifhashemi, A & Nikgohar, F 2005, 'MILO: Personal Robot Platform', paper presented to IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada.

Savage Innovations 2005, *OOPic ® The Hardware Object (tm)*, viewed 10 March 2005, <http://www.oopic.com/>.

Sharp Electronics Corp. *GP2D12 Sales & Technical Information*, viewed 30 October 2005, <http://www.sharpsma.com/part.php?PartID=3900>.

Sprouse, J *nomadic.sourceforge.net*, viewed 30 October 2005, <http://nomadic.sourceforge.net/>.

Thrun, S, Burgard, W & Fox, D 2000, 'A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping', paper presented to IEEE International Conference on Robotics and Automation, San Francisco, CA, 24-28 April 2000.

Vamplew, P 2004, 'Lego Mindstorms Robots as a Platform for Teaching Reinforcement Learning', paper presented to Proceedings AISAT2004: International Conference on Artificial Intelligence in Science and Technology, Hobart, Tasmania, Australia.

Wagner, B, Hohmann, P, Gerecke, U & Brenneke, C 2004, 'Technical Framework for Robot Platforms in Education', paper presented to International Conference on Engineering Education and Research, Olomouc and Bouzov Castle, Czech Republic.

Will, JD, Stombaugh, TS, Benson, ER, Noguchi, N & Reid, JF 1998, 'Development of a Flexible Platform for Agricultural Automatic Guidance Research', paper presented to ASAE Meeting, ASAE 2950 Niles Road, St Joseph, MI 49085-9659 USA.

Zagros Robotics, viewed 18 May 2005,
<https://www.zagrosrobotics.com/Index.asp>.

Appendix A Straight Line Test Data

Run 1:

Experiment A				
Trial No:	Hit Wall:	X Back Travel (mm)	Y Back Travel (mm)	Y Front Travel (mm)
1	N	5582	515	475
2	N	5579	704	679
3	Y	5498	208	181
4	N	5569	607	616
5	Y	4992	225	177
6	N	5521	1163	1197
7	N	5518	1258	1303
8	Y	5358	202	183
9	N	5513	904	900
10	N	5500	411	388
11	Y	4847	234	180
12	Y	3222	279	172
13	N	5548	811	804
14	N	5551	709	677
15	N	5585	568	548
16	Y	4822	206	179
17	N	5564	848	853
18	N	5528	381	311
19	N	5569	622	590
20	N	5569	670	654
21	Y	2449	231	179
22	N	5577	309	295
23	Y	4628	228	177
24	N	5549	470	451

Experiment B				
Trial No:	Hit Wall:	X Back Travel (mm)	Y Back Travel (mm)	Y Front Travel (mm)
1	N	4772	586	574
2	N	4771	561	574
3	N	4812	567	535
4	N	4814	345	270
5	N	4789	918	919
6	N	4916	784	734
7	N	4934	355	300
8	N	4862	955	976
9	N	4781	610	577
10	Y	4704	257	193
11	N	4985	530	478
12	N	4664	1113	1142
13	N	4792	1023	1120
14	N	4728	398	360
15	Y	5509	919	844
16	Y	4722	773	738
17	N	4597	679	646
18	Y	4538	279	185

19	N	4718	695	660
20	N	4874	768	729
21	N	4795	575	490
22	Y	4804	264	185
23	N	4745	1002	906
24	N	4873	476	425

Experiment C				
Trial No:	**Hit Wall:**	**X Back Travel (mm)**	**Y Back Travel (mm)**	**Y Front Travel (mm)**
1	N	4438	625	616
2	N	4418	637	602
3	N	4310	767	783
4	N	4244	645	817
5	N	4321	686	590
6	N	4262	822	850
7	N	4247	613	557
8	N	4261	876	969
9	N	4343	848	877
10	N	4303	779	740
11	N	4352	813	753
12	N	4325	701	649
13	N	4237	698	788
14	N	4321	871	804
15	N	4289	587	544
16	N	4307	709	802
17	N	4304	855	848
18	N	4316	838	809
19	N	4389	520	524
20	N	4298	640	603
21	N	4264	692	660
22	N	4187	794	813
23	N	4290	780	727
24	N	4262	858	866

Run 2:

Experiment A				
Trial No:	Hit Wall:	X Back Travel (mm)	Y Back Travel (mm)	Y Front Travel (mm)
1	N	4954	750	754
2	N	4950	753	758
3	N	4946	795	799
4	N	4954	739	740
5	N	4944	767	771
6	N	4945	762	768
7	N	4952	748	749
8	N	4954	799	806
9	N	4953	759	764
10	N	4958	745	748
11	N	4958	717	720
12	N	4956	734	733
13	N	4949	695	702
14	N	4967	680	679
15	N	4956	727	730
16	N	4951	728	736
17	N	4956	738	740
18	N	4943	767	770
19	N	4958	704	709
20	N	4953	717	718
21	N	4948	778	782
22	N	4942	792	802
23	N	4959	760	763
24	N	4953	764	768

Experiment B				
Trial No:	Hit Wall:	X Back Travel (mm)	Y Back Travel (mm)	Y Front Travel (mm)
1	N	4945	817	819
2	N	4945	1032	1065
3	N	4956	545	534
4	N	4946	820	830
5	N	4952	771	769
6	N	4950	802	804
7	N	4944	830	844
8	N	4942	850	846
9	N	4949	781	787
10	N	4963	566	561
11	N	4959	671	670
12	N	4959	704	701
13	N	4949	702	762
14	N	4958	666	670
15	N	4941	869	882
16	N	4960	574	562
17	N	4951	700	709
18	N	4948	738	742
19	N	4964	596	588
20	N	4950	706	705

21	N	4949	692	691
22	N	4955	630	614
23	N	4963	584	560
24	N	4945	772	775

Experiment C				
Trial No:	**Hit Wall:**	**X Back Travel (mm)**	**Y Back Travel (mm)**	**Y Front Travel (mm)**
1	N	4983	636	665
2	N	4982	652	542
3	N	4958	779	743
4	N	4986	727	749
5	N	4970	764	725
6	N	4981	692	722
7	N	4952	633	656
8	N	4963	658	644
9	N	4961	778	755
10	N	4980	685	704
11	N	4966	735	683
12	N	4983	769	750
13	N	4953	776	745
14	N	4968	679	689
15	N	4968	706	739
16	N	4986	754	741
17	N	4980	767	759
18	N	4967	701	708
19	N	4972	744	713
20	N	4971	728	679
21	N	4958	660	606
22	N	4969	664	679
23	N	4966	707	686
24	N	4970	672	638

Appendix B UMBmark Data

Absolute Position		Odometry Computed Position		Odometry Position Errors	
X	Y	X	Y	X	Y
Clockwise					
1301	503	811.1987464	536.4523522	489.8013	-33.4524
1347	505	808.5313349	571.3640254	538.4687	-66.364
1478.5	379	804.5748239	584.2257465	673.9252	-205.226
1373	366.5	842.5614822	538.3531557	530.4385	-171.853
1377	430	849.0291704	538.4963574	527.9708	-108.496
Counter Clockwise					
1468	506	555.8567847	817.7140851	912.1432	-311.714
1142	785.5	553.2927637	821.3746561	588.7072	-35.8747
846	879	536.2154631	818.5114477	309.7845	60.48855
1089.5	878	550.304378	822.2886719	539.1956	55.71133
1266.5	682.5	570.7846037	782.6891346	695.7154	-100.189

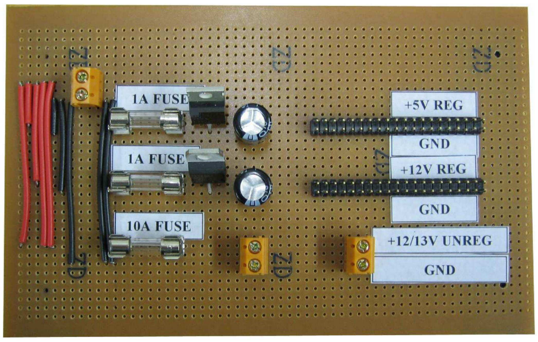
Centres of Gravity		
	X	Y
CW	552.1209	-117.078
CCW	609.1092	-66.3156
Absolute Offset of Centres of Gravity		
CW	564.3977	
CCW	612.7086	
$E_{\max.syst}$ (UMBmark)		
612.7086		

Appendix C Mobile Robot Motion Capabilities

	Turning Circle			X Axis		Yaxis		Z Axis				
Platform Type	X	Y	Z	Fwd	Rvs	Left	Right	Up	Down	Drive System	Date Accessed	Reference/url
Terrestrial:												
Flexible Robot Platform	N	N	0	Y	Y	N	N	N	N	Differential		
HandyCar	N	N	>0	Y	Y	N	N	N	N	Rear wheel Drive	30/10/2005	Linder 2001
ActivMedia Pioneer 1	N	N	0	Y	Y	N	N	N	N	Differential	30/10/2005	http://www.activrobots.com/
Evolution Robotic ER1	N	N	0	Y	Y	N	N	N	N	Differential	30/10/2005	http://www.evolution.com/er1/
K-Team Khepera II	N	N	0	Y	Y	N	N	N	N	Differential	30/10/2005	http://www.k-team.com/index2.html
Nomad Super Scount II	N	N	0	Y	Y	N	N	N	N	Differential	30/10/2005	Maxwell & Meeden 2000
Micro Robot Alice	N	N	0	Y	Y	N	N	N	N	Differential	30/10/2005	http://www.k-team.com/index2.html
VolksBot	N	N	0	Y	Y	Y	Y	N	N	Omnidirectional	30/10/2005	http://www.intelligentcomponents.com/abheek/docs/fne11.pdf
Aquatic - Underwater:												
CETUS™ AUV	0	0	0	Y	Y	N	N	Y	Y	Differential Thrust	30/10/2005	http://auvlab.mit.edu/vehicles/vehiclespecCETUS.html
Autosub	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Direct Drive	30/10/2005	http://www.soc.soton.ac.uk/OED/index.php?page=as
Ocean Voyager AUV	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://www.hboi.edu/eng/auv_systems.html
OCEAN VOYAGER II	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://www.oe.fau.edu/AMS/auv.html
R1 Project	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://underwater.iis.u-tokyo.ac.jp/robot/r1/r1-chp1-e.html
REMUS - Hydriod Inc	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Direct Drive	30/10/2005	http://www.hydroidinc.com/remus.htm#Specifications:
Fetch 3.5 - Sias Patterson Inc	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://www.siaspatterson.com/
MARIDAN 600	>0	>0	>0	Y	Y	N	N	N	N	Differential Thrust	30/10/2005	http://www.maridan.dk/product/productspecification.html
Aries AUV	>0	0	0	Y	Y	Y	Y	Y	Y	Differential Thrust & Thrusters	30/10/2005	http://www.cs.nps.navy.mil/research/auv/auvstats.html#physical
MicroSeeker	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://www.huv.com/uSeeker/index.html
Theseus AUV - ISE Research Ltd.	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://www.ise.bc.ca/theseus.html
Kambara - Aus Nat Uni	0	0	0	Y	Y	N	N	Y	Y		30/10/2005	http://users.rsise.anu.edu.au/~kambara/2-Systems/Submersibles.html
ONR Hovering AUV	0	0	0	Y	Y	Y	Y	Y	Y	Multiple Thrusters	30/10/2005	http://auvlab.mit.edu/vehicles/vehiclespecONRHAUV.html
Odyssey Iix AUV	>0	>0	>0	Y	Y	N	N	N	N	Single Prop Drive	30/10/2005	http://auvlab.mit.edu/vehicles/vehiclespec2x.html
Twin Burger AUV	N	N	0	Y	Y	Y	Y	Y	Y	Differential Thrust & Thrusters	30/10/2005	http://underwater.iis.u-tokyo.ac.jp/robot/tb/tb-chp1-e.html
RAUVER Mk2	N	N	0	Y	Y	Y	Y	Y	Y	Differential Thrus & Thrusters	30/10/2005	http://www.ece.eps.hw.ac.uk/~kelvin/oceansweb/rauver/rauver_main_mk2.htm
Aquatic - Surface:												
AutoCat Autonomous Surface Craft	N	N	>0	Y	Y	N	N	N	N	DC Motor or Outboard	30/10/2005	http://auvlab.mit.edu/vehicles/vehiclespecASC.html#
Airborne - Plane:												
Generic Aeroplane	>0	>0	>0	Y	N	N	N	N	N	Single Prop Pull		
UAV Ariel	>0	>0	>0	Y	N	N	N	N	N	Single Prop Pusher	30/10/2005	http://www.aeromech.usyd.edu.au/wwwdocs/ariel.html
UAV Brumby Mk. II	>0	>0	>0	Y	N	N	N	N	N	Single Prop Pusher	30/10/2005	http://www.aeromech.usyd.edu.au/wwwuav/uav_brumby_intro.html
UAV T-Wing - Standard flight	>0	>0	>0	Y	N	N	N	N	N	Twin Prop Pull	30/10/2005	http://www.aeromech.usyd.edu.au/uav/twing/
- Vertical flight	>0	>0	0	Y	Y	Y	Y	Y	Y			
Airborne - Helicopter:												
Ursa Magna series	>0	>0	0	Y	Y	Y	Y	Y	Y	Main & Tail Rotor	30/10/2005	http://robotics.eecs.berkeley.edu/bear/testbeds.html

Appendix D Power Distribution Board Specifications

FRP Power Distribution Board



Overview

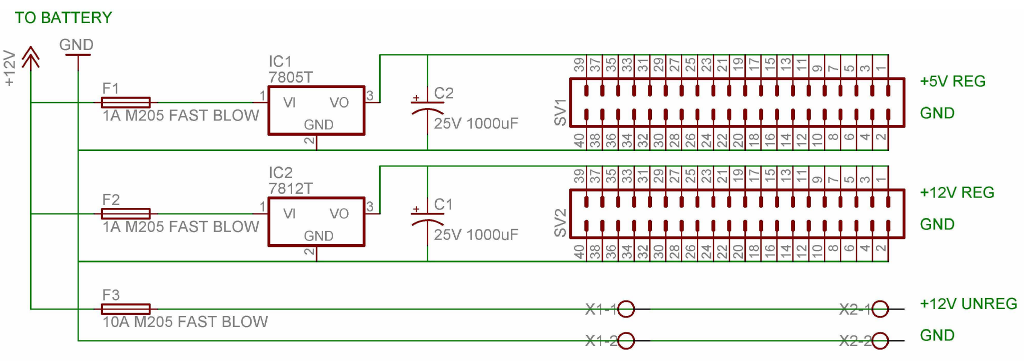
The FRP Power Distribution Board is designed to supply power to the components of the University of Tasmania Flexible Robot Platform. The Board takes power from any 12V power source and provides the following connections:

- Regulated 5V 1A Max Load
- Regulated 12V 1A Max Load
- Unregulated 12V 10A Max Load

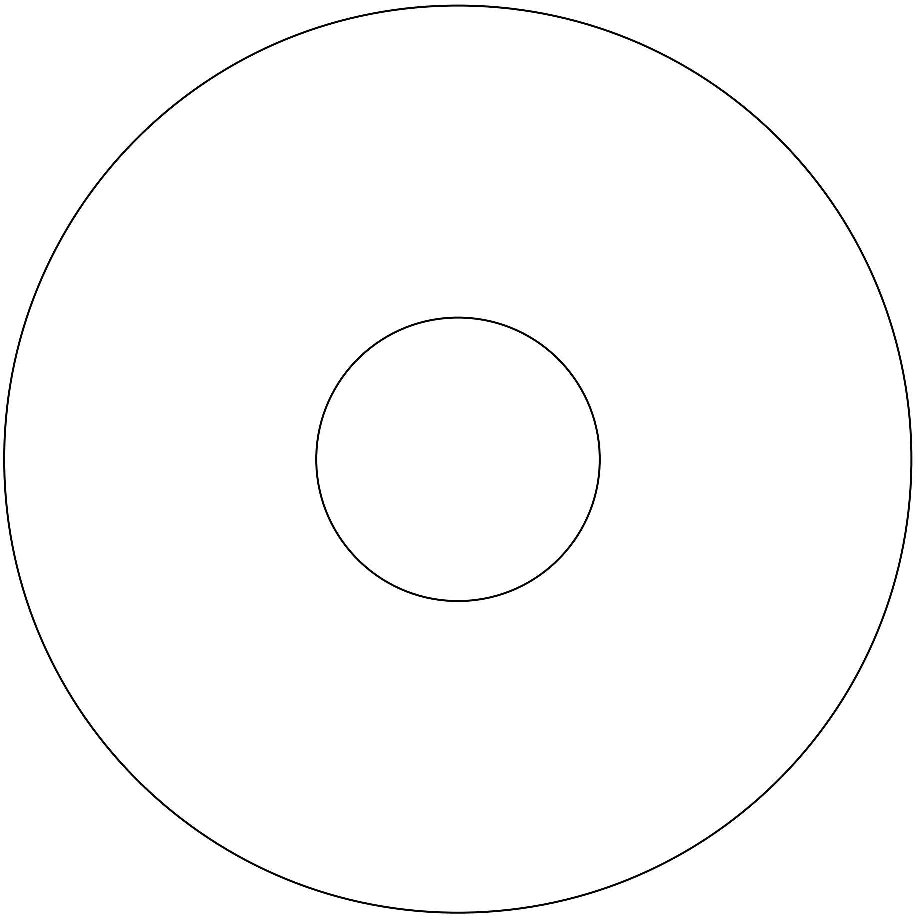
The two regulated supplies are for sensitive components on the FRP and the unregulated supply is for the FRP Motors and other high current devices. The capacitors on the regulated supply lines prevent voltage drops from the operation of high current components (motors). All of the supplies are fused to protect FRP components.

Note: Take care to plug a component into the correct voltage supply to prevent damage.

Schematic



Appendix E Compact Disc



Appendix F OOPic API Documentation